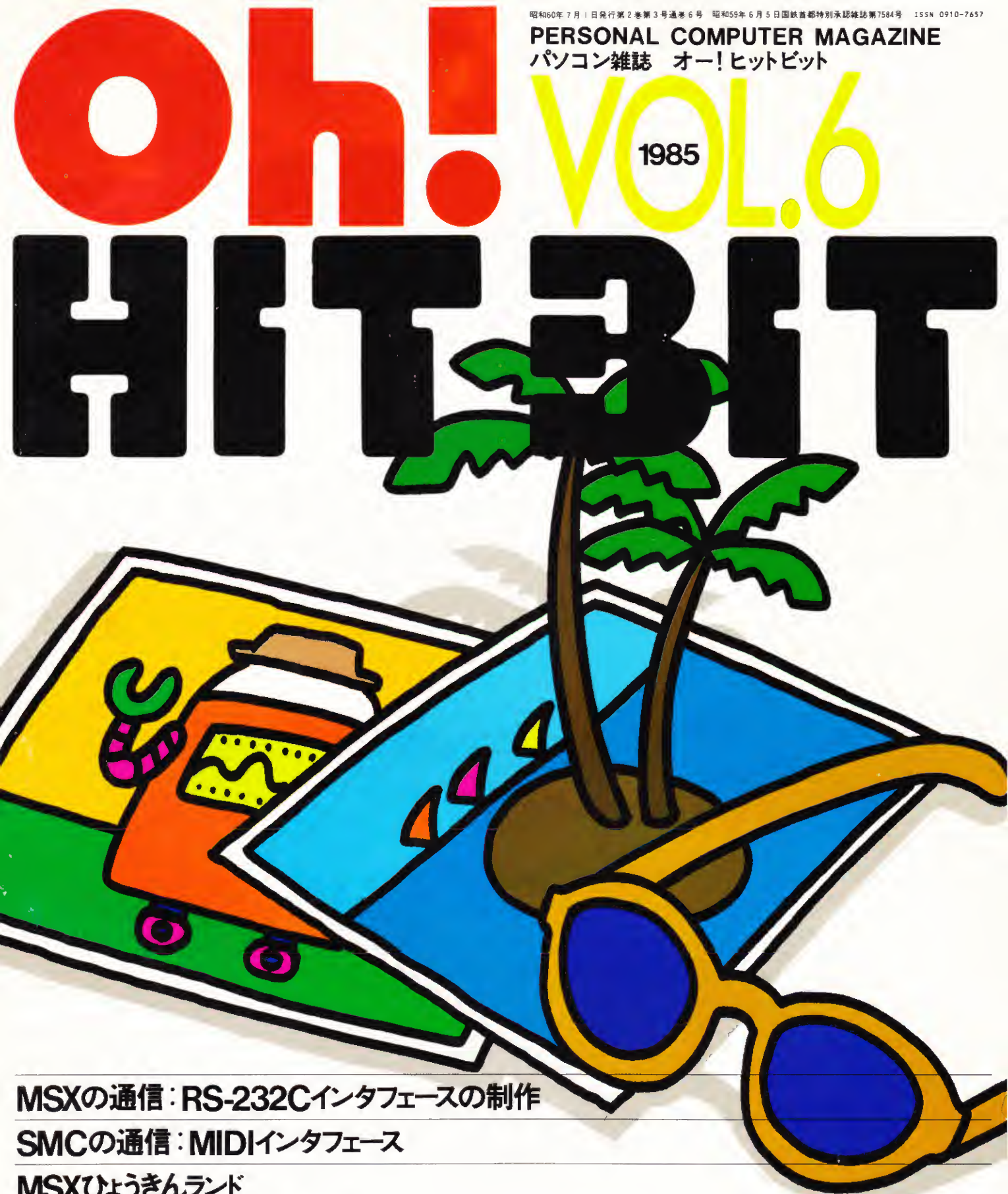


PERSONAL COMPUTER MAGAZINE  
パソコン雑誌 オー! ヒットビット



MSXの通信：RS-232Cインタフェースの制作

SMCの通信：MIDIインタフェース

MSXひょうきんランド

SMCリンクパッケージ集vol.2

MSXミュージックエディタII、サウンドエディタ

SMCオリジナルゲーム「AREA-X」

定価480円

SONY

64KB

僕は64Kバイトの自由を手に入れた。





# メッツォーの仲間がふえた。

RAM64KバイトのメッツォーHB-201新登場。  
人気のメッツォーといっしょに、MSXをさらに面白くする。

RAM64KBの本格派HB-201と、16KBの  
人気者HB-101。楽しみ広げる2スロット搭載。  
メッツォーの新しい仲間、HB-201新登場。RAM  
64KBの実力で複雑なプログラムやゲームにも余  
裕で対応。ソフトもさらに幅広く楽しめます。また、  
人気のHB-101はスタンダードなRAM16  
KB。発売のエクステンションメモリーカートリッジ  
装着でRAM64KBまで拡張可能。気軽に楽し  
めるパソコンです。もちろん、どちらも2スロット搭載。

フロッピーディスクドライブと組んで、よりダイナミ  
クなプレイも行えます。データレコーダやカラープ  
ロッタープリンタなどの周辺機器と接続すれば  
楽しみが倍増すること受けあいです。これからは、  
ふたつのメッツォーがMSXをますます面白くします。  
カーソルジョイスティックやポーズボタン付で  
プレイが面白い。便利なキャリングハンドル付。  
エキサイティングにゲームが楽しめるカーソルジョ  
イスティックやゲームを一時中断できるポーズボタ

ン。持ち運びに便利なキャリングハンドル。など、  
メッツォーはみんなにうれしい美しいパソコンです。  
住所録などが簡単につくれるHIT BITノート  
機能内蔵。16色カラー表示。サウンド機能。  
ソニー独自のHIT BITノート機能内蔵。住所録、  
スケジュールなどが誰でも簡単につくれます。また、  
MSX規格だからソフトも充実。もちろん16色のカ  
ラー表示や8オクターブ3重和  
音のサウンド機能も付いています。

**MSX**

# HIT BIT MEZZO

新登場 **64KB**



新登場。基本機能に、さらにRAM64Kバイト搭載。  
これは、余裕の人のメッツォー。

**HB-201** ソニーパーソナルコンピュータHB-201  
¥59,800  
ボディカラーはツヤ消しのブラックとライトグレーの2タイプです。

美しいフォルムが人気のパソコン。基本機能に徹した、  
これは、夢中の人のメッツォー。

ソニーパーソナルコンピュータHB-101  
¥46,800 **HB-101**  
ボディカラーはレッド、ブラック、アイボリーの3タイプです。

●カタログを差しあげます。ハガキに、住所・氏名・年齢・職業・機種名をご記入のうえ、お申し込みください。〒108東京都高輪局区内ソニー株式会社営業本部カタログ「O・H」係 ●MSXはマイクロソフト社の商標です。



# SONY

## ひとりで、8言

様々な高級言語を3.5インチフロッピーで供給。

CP/Mマシンだから、楽しみは多彩だ。

パソコンを深く知れば知るほど、単にアプリケーションソフトを利用するだけでなく、自分自身で様々な言語を使ってプログラミングを楽しみたいくなります。ソニーのSMC-777Cは、言語の面白さを追求できるパソコン。CP/Mディスク

(別売)を搭載すると、システム全体が

高性能CP/MマシンになりFORTRAN, FORTH, COBOL, PROLOG, Cなど各種高級言語を走らすことができます。しかも、SMC-777C用に数多くの高級言語を3.5インチフロッピーで用意。言語のもつ可能性を、あなたの手で確かめることができます。

### ■SMC-FORTRAN

世界で最初に開発された高級言語です。SMCでは数値計算に適し、科学技術計算や機器の制御に優れたANSI-FORTRANのサブセットを採用しています。(型名SMJ-S037D ¥14,800)

### ■SMC-COBOL

事務用共通言語。SMCでは、世界の標準となるANSI-74がベースです。非常にコンパクトで、48 K RAMシステムでも約4000ステートメントのプログラム実行可能。(型名SMJ-S038D ¥14,800)

### ■SMC-PASCAL

科学技術計算、事務処理、コンピュータ教育、アルゴリズム記述などに適した言語。1960年代





# 語。ザ・8ビット。

元にもって設計されました。プログラミング言語を系統的に学ぶことができ、初めての人でも容易に理解できます。(型名SMJ-S039D ¥14,800)

■SMC-C  
32ビットとしてベストセラーの実績をもつ「BDS-C」。構造化プログラミング、移殖性、オブジェクトの書き込みなどで注目されている言語です。短期間に学ぶことができ、しかも本格的なプログラミングにも適しています。(型名SMJ-S040D ¥29,800)

■LISP 80  
人工知能言語として、各方面から注目されている言語。歴史は古く、FORTRANとはほぼ同時期の1960年に発表されています。専門的な問題解決や推論、自然言語の処理、知的支援システム、などに向いています。(型名SMJ-S041D ¥18,500)

■SMC-FORTH  
チャールズ・ムーアが開発した自己増殖型、スタック・オリエンテッド型の言語。目的によりどの分野にも適しています。OS、テキストエディター、アセンブラデバッグが内蔵。(型名SMJ-S053D ¥14,800)

■SMC-PROLOG  
「第5世代コンピュータ計画」の核言語モデルに採用され、人工知能言語／知識処理言語として注目されている論理型プログラム言語。1972年フランスで誕生。(型名SMJ-S054D ¥14,800)

■SMC EDIT with CP/M-80  
20種類のコマンドと豊富なカーソル移動機能によってプログラムの作成・修正を簡単にしたスクリーンエディターです。(型名SMJ-S036D ¥14,800)

が8ビットの実力。付属の3.5インチディスクには、Dr.Logoや簡易言語、アセンブラもある。

SMC-777Cには、3.5インチのオリジナルディスクが付属しています。このディスクは、ソニー独自の777-BASICや次世代の言語Dr.Logo、本格的アセンブラ777-ASSEMBLER、機械語デバッグ777-DEBUGGER、簡易言語777-MEMOなど8種類のソフトを装備。特に、1967年マサチューセッツ工科大学の人工知能研究所で開発されたDr.Logoは、独自のタートルを簡単なコマンドとメッセージだけで動かし、新しい可能性を創造できる言語として注目されています。また、これらのソフトは3.5インチのドライブから呼び

込まれ、Sony FILERというCP/Mバージョン1.4と互換性をもつOSのもとで作動するようになっています。このOSは、ユーザーにも開放されています。

■4096色から16色を。  
SMC-777Cは、内蔵のカラーパレットにより4096色から16色を表示可能。単色を選んでもその16輝度表示ができるため、ミニコンなみの3次元グラフィックを描けます。さらに付属のグラフィックディスクを用いれば、図形処理もいっただけ簡単にできます。その他、ミュージックディスクも付属しているの

**HIT 3IT**

SMC FORTRAN, COBOL, PASCAL, C, FORTH, PROLOG, EDITは発売元ソニー・コンピュータ・エンタープライズ設計。●CP/M, Dr.Logoはデジタルリサーチ社の商標。●カタログを差し上げます。ハガキに、住所・氏名・年齢・職業・機種名をご記入のうえ、お申し送りください。〒108 東京都港区三田4-1-1ソニー・コンピュータ・エンタープライズ株式会社。

新しい世界を、発見できそうだ。  
**SMC-777C**

下の写真は、ソニー・パーソナルコンピュータSMC-777C ¥168,000(本体)とモニター・カラーディスプレイCPD-14CDI ¥73,800/ケーブルSMK-702 ¥6,800とのセットです。

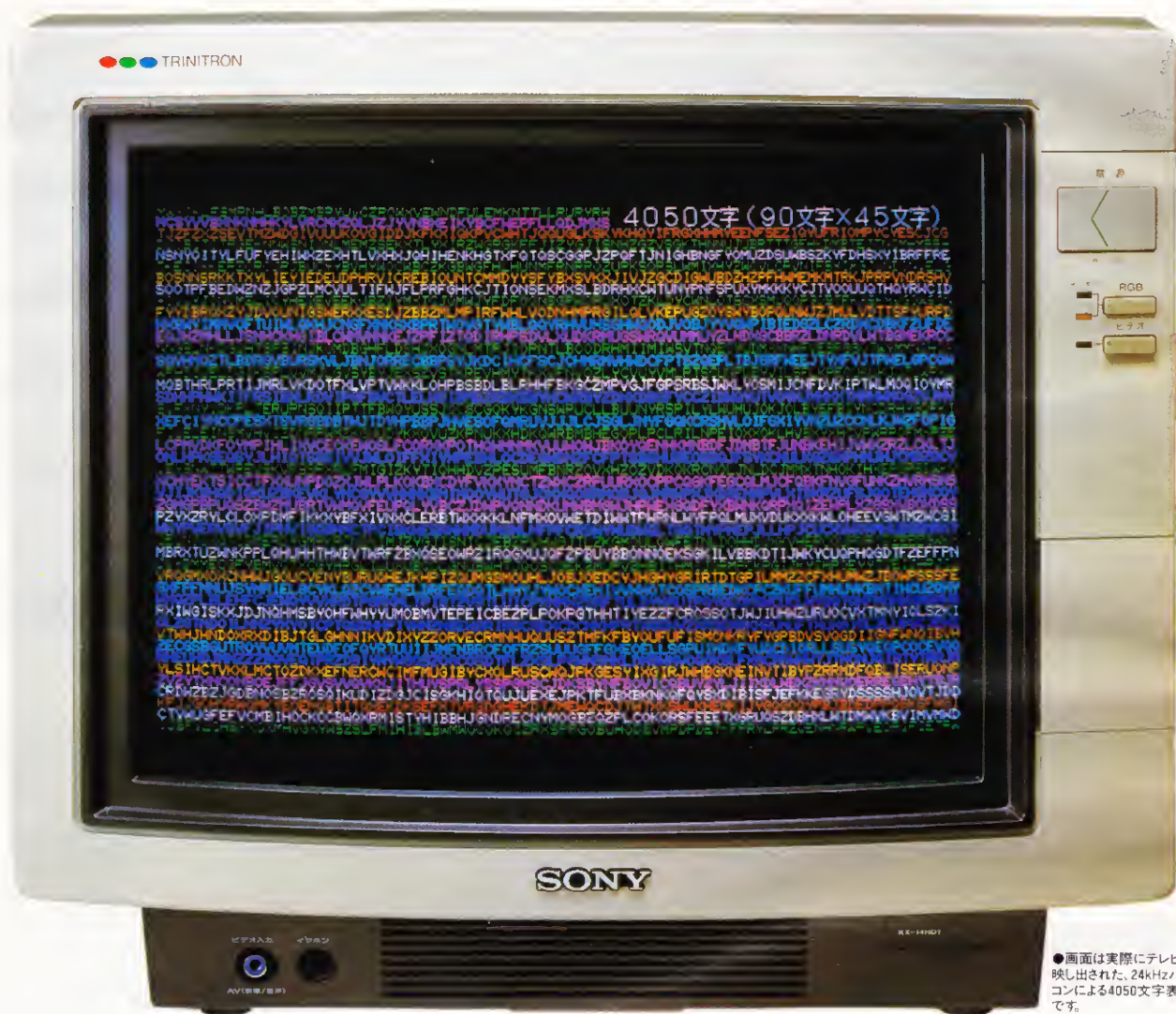


ソニー・パーソナルコンピュータ  
**SMC-777C**  
¥168,000

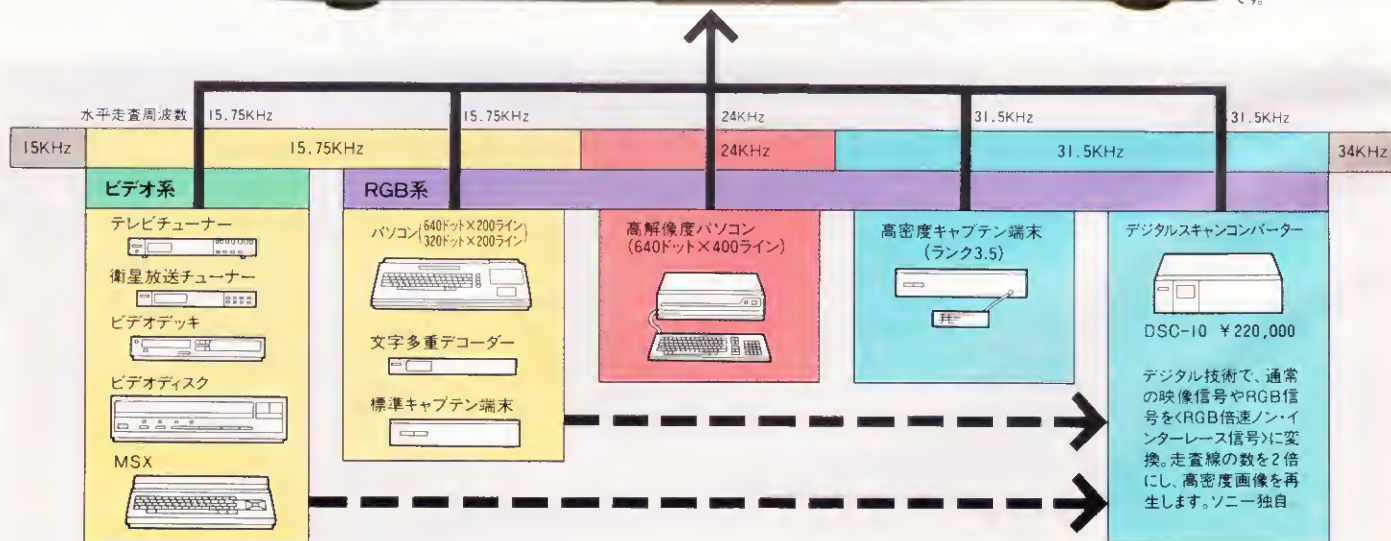


# SONY

## 映像ソースに合わせてマ トリニトロンカ



●画面は実際にテレビに  
映し出された、24kHzパソ  
コンによる4050文字表示  
です。





# マルチスキャン(水平走査周波数15kHz~34kHz)する。 カラーモニター。

## MULTISCAN



ビデオから高解像度パソコン、高密度キャプテンまで、これ1台でマルチに対応します。マルチスキャン機能搭載。

●**新開発マルチスキャン機能**。水平走査周波数15kHzから34kHz、垂直走査周波数50Hzから60Hzまで、広範囲な走査周波数に連続して自動追従する、画期的な新機能です。テレビ、ビデオから高解像度パソコン、高密度キャプテンまで映像ソースの進化に、これ1台でマルチに対応できます。●**豊富な入出力端子群**。RGB系ニューメディアソースをつなげる、アナログ21ピン/デジタル8ピン2系統。加えて、ビデオ系MSXパソコンやビデオ、TVゲームなどを1本のコードで前面から接続できるAV Hit端子も装備。●**ファインブラックトリニオン管**。0.37mmファインピッチ管により、文字・図形情報はシャープに、映像はきめ細かな美しさで再現。マルチスキャン機能によって2000文字(80文字×25文字)も4050文字(90文字×45文字)も、鮮明に映し出します。

マルチスキャントリニオンカラーモニター  
14型KX-14HD1.....¥125,000

別売AV Hit端子用オーディオ・ビデオケーブル.....●VMC-610S ¥1,500  
(ステレオ用) ●VMC-611MS ¥1,500(モノラル用) ●VMC-612MS ¥1,700(モノラル用)

※2000信号の場合、マニュアル調整が必要です。

# BLACK TRINITRON

(2000文字表示高画質  
ディスプレイテレビ)

0.37mmファインピッチ、2000文字対応、RGB入力(21ピン/8ピン)、前面AV Hit、ビデオ入力端子装備の高画質テレビ。■カラーテレビ  
KV-14CP1 ¥99,800

(パソコン専用、2000文字対応ディスプレイ)

0.37mmファインピッチ、最大2000文字を鮮明に表示。RGB入力(アナログ21ピン/デジタル8ピン)装備。■キャラクターディスプレイ  
CPD-14CD1 ¥79,800



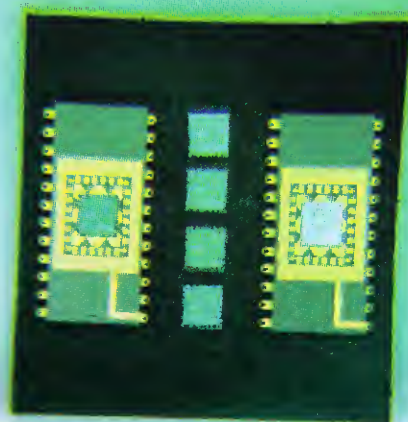
●カタログ請求は、住所・氏名・年令・職業・電話番号・機種名を明記の上、〒108東京都高輪区区内ソニー株式会社カタログ係へハガキで、お申し込みください。





# Microcomputer Show '85





## マイクロコンピュータショウ 85

5月22日水～25日土)にかけて、東京平和島の東京流通センターにおいてマイクロコンピュータショウ'85が開催された。これは、日本電子工業振興協会が主催して例年開催されているもので、毎回話題の新製品が発表されることで有名である。今年もMSX2の発表、16ビットパソコンの新製品など、かすかすの話題を集めた。Oh! HITBIT取材班もさっそく取材してきたので報告しよう。

Sonyのブースは第1会場の中央、NECのブースのとなりにあり、SMCシリーズ、MSX、そして話題の新製品、パーソナルワープロ HITBIT-Wordなどを中心に展示をしていた。

SMC関係の展示では、RS-232Cを使った通信関係のデモを走らせていた。今年は通信の自由化や、郵政省方式のパソコン間通信方式の発表もあり、ますます通信に対する関心が高まると思われる中、タイムリーな展示といえるであろう。

MSXではHBシリーズの最新型、HB-201の発表がおこなわれていた。これは、HB-101の64KRAMバージョンで、ライトグレーとつや消しブラックのボディーカラーを採用し、高級感あふれる高性能機である。

MSXといえばこのショウの話題の一つに、MSX 2があるが、Sony ブースにはその姿は見えなかった。MSXでトップシェアを得たHBシリーズのMSX2バージョンはいつ登場するのだろうか。

さて、このブースで一番大きな面積を占めていたのが、HITBIT-Wordである。これは、Sony が一般向けに開発したパーソナルワープロで、漢字3000字を使用可能、5色印字可能な熱転写プリンタを装備して98,000円と、手軽に使えるワープロになっている。ブースではこのHITBIT-Wordをたくさんならべて、来場者に自由に操作させていた。

そのほか、COMDEXではじめて姿をあらわしたVIEW SYSTEMが、マイコンショウ





でも展示されていた。このVIEWの中心は16ビットパーソナルコンピュータの、SMC-2000である。CPUに80186を採用、640×400、256色と魅力的なマシンであるが、まずは北米のみでの発売で、日本での発売は未定だそうである。

それでは、Sony以外のブースに目を転じてみよう。まずはMSX2の動向である。MSXとの互換性を保ちながら、メモリを最大128Kバイトにまで拡張可能にし、グラフィックスの解像度を最大512×212ドットに拡張、キャラクタ画面で80文字×24行の表示が可能、スプライト、クロック機能の強化、FM音源の採用など機能を大幅に強化したものである。すでに発表し、発売したのがヤ

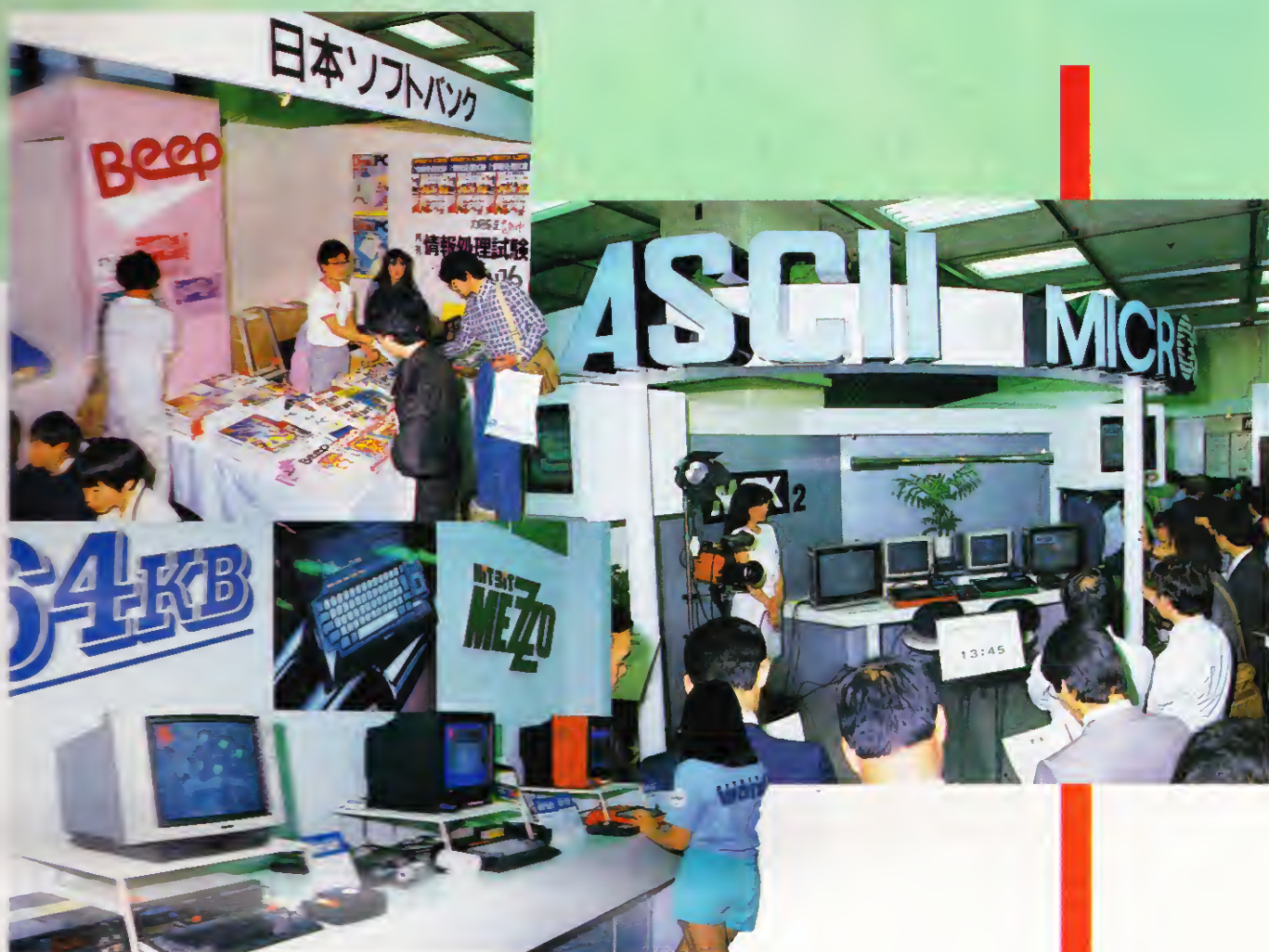
マハと東芝で、松下、三洋、三菱などが参考出品の形で試作品を展示していた。MSX2用のVDPをアスキーと協同開発したヤマハと、MSXエンジンなどの半導体製品をつくっている東芝が実際の商品開発でも先行したのは当然であろう。Sonyも得意のAV技術を生かした特色ある製品を開発してもらいたいものである。

それ以外の新製品といえば、なんといっても、NECの新16ビットマシン、PC98XAとPC9801U2であろう。PC98XAはCPUが、80286、メインメモリ512Kバイト、グラフィックスは1120×750ドット、4096色中16色、24×24ドットの明朝体の漢字フォントなど、どの点をとっても最高の16ビットマシンに仕上がっている。会場では、陰面消去した

多面体のアニメーションのデモを行っていたが、そのスピードはすばらしかった。PC9801U2は、PC9801のトランスポート版で、CPUに日電オリジナルのV30を初めて採用、4096色中8色(拡張時は16色)、3.5インチフロッピーを内蔵、小型軽量になったタイプである。また、640×400ドット対応のプラズマディスプレイも用意されている。NECでは、ホームパソコンPC-6000シリーズでは3.5インチを採用していたが、今回のPC-9800シリーズでの採用で、主要メーカーのほとんどが3.5インチを採用したことになり、次世代のフロッピーディスクの地位を確固たるものとした。

富士通のブースではFM16 $\pi$ 、FM16 $\beta$ 、9450





2000機種が目立った。とくに9450Σのマルチウインドウ画面には感動したが、これは完全にビジネス用でホビイストが使う機種ではない。NECが80286を採用し、富士通はどう対応するであろうか。

シャープのブースではX1シリーズの新製品が展示されていた。X1ターボ Mode 100、そしてX1の普及機種X1F、Mode 110、20である。X1ターボMode 140は、システム、ユーザ辞書などにより漢字処理を強化したもので、X1F、Mode 110、20は、X1をさらに低価格化し、NEW-BASICにより、機能はX1ターボなみに向上させたものである。

シャープのブースでは、マッキントッシュ用のかな漢字入力ソフト、EG-BRIDGEが

展示されていた。マッキントッシュの日本語化作業は遅れている模様だが、ソフトにより実現してしまったわけだ。しかし、このようにソフトでサポートする方式では使えるソフトが限られてしまう。マッキントッシュのような使い易いパソコンが、日本語入力の問題だけで日本で売れないのは残念である。一日もはやく、マッキントッシュの漢字版を開発してもらいたい。

その他、目立ったブースにハドソンのブースがあった。ここは、アメリカ西部のイメージの丸太小屋をかたどっており、さまざまな機種用のゲームを展示していた。また、ローランドのブースでは、シンセサイザをMIDIインタフェースで自動演奏するシステムを展示していた。今月号の記事でSMC

にもMIDIインタフェースがつながったことだし、SMC用のMIDI用ソフトも作ってほしいものだ。

マイコンショウとおなじ時期に、晴海の国際貿易センターにおいて、第60回ビジネスショウも開催された。去年から、マイコンショウと重なったので片方しか出展しない企業も多かった。Sonyはビジネスショウには参加していない。Sonyの文書管理システムSIOSが見られるかとおもったが、残念である。





# RAID ON BUNGELING BAY

●Imokichi Himizu



## MSX新作ゲーム紹介

### ●プロローグ

夏だというのにヤケに肌寒い今日、オレは旅を決意した。目的などない。ただ、このゲームまみれの日常から脱出したかっただけだ。

人生の転機は突然やってくる。そして今がそのときなのだ、と、自分に納得させつつ、オレは駅までの道を急いだ。別に急ぐ旅ではないのだが、早くしないと、日常という名の泥沼から抜け出せなくなってしまう、という強迫観念が、オレの足を早めさせたのだろう。

気が付くと、そこはひなびた温泉町の、1軒の旅館の前だった。宿帳に記入をしたオレは、なにをおいてもまず、畳の上で大の字になって、ねそべることにした。くつろぎ、そう、これこそがオレに欠けていたものだったんだ。ここには急がず編集者も、怒る編集者もない。嗚呼、極楽//

と、そこへ、旅館の女中がやってきた。「お客様、このゲームカートリッジを差し上げますわ。」

「あつ、これは、まだ未発表のはずの、MSX版バンゲリングベイじゃないか。どうしてこれを僕に……？」

「だって、そのトレーナ……。」

なんとオレは、ゲーム雑誌『Beep』のトレーナを着ているじゃないか！

やっぱりオレは、ゲームからは足を洗えない運命になっているんだ。ええい、こんな



つたら、このカートリッジを、オー・ヒットビットの編集者のNさんのところへ持って行って、1本記事を書くっきゃない！

N「まずは自己紹介してもらいましょう。」

芋「はい。えっと、僕はBeepのライターやってる、氷水芋吉です。Beepの方で、ファミコン版のバンゲリングベイのレビューをやった関係上、今回のMSX版バンゲリングベイの方にかりだされました。こんなところでいいですかね。」

N「けっこう、けっこう。けっこうを見ずに日光というなかれですよ。それから、私が編集者のNだす。どうぞよろしゅう。」

芋「実は、僕のペンネームの氷水芋吉っていうのも、このNさんの命名なんですけれど、まあ、そんなことはどーでもいいか。えーっと、バンゲリングベイの話でしたよね。」

N「つまり、今回ソニーから、バンゲリングベイが発売になるわけね。発売日は1月後の7月21日なんだけど、そこは編集者の

親心、ヒットビットの読者には特別に教えちゃおうってわけ。でも、ただ紹介するだけなら猿にだってできますよ。んでもって、芋吉つつあんには、Beepのライターとしての名誉を賭けてもらって、バンゲリングベイについて隅から隅までしゃべってもらおう、という具合になっているんだな、これがまた。そんなわけだから、まずはバンゲリングベイの生い立ちでもいってもらおうかな。ほれ言え早よ言え今言え直ぐ言え。」芋「えーっと、そもそもバンゲリングベイっていうのは、アメリカのコンピュータ用に、かの有名なブローダーボンド、あのロードランナーの会社なんだけど、そこからリリースされているゲームなんです。そして、このMSXのバンゲリングベイっていうのは、ソニーお得意の移植版で、実はファミコンのバンゲリングベイよりも本物そっくりだ、という噂もあるぐらいの、スグレモノなんです。」

N「それでゲームの内容はどんなものなのかな？」

芋「えっ？ Nさんはまだやってないんですか？」

N「やった者の負けですよ。」

芋「じゃあ僕が説明します。まず、プレイヤーは、ジェットヘリに乗ってバンゲリング帝国へ戦争しにいくわけですよ。そのヘリコプターには、爆弾9発とバルカン砲が搭載されていて、その爆弾で、バンゲリング帝国の工場を破壊し、爆破活動を邪





## ●エビローグ

オレは今まで、ゲームだらけの日常に飽き飽きしていたというより、ただ単に面白いゲームに餓えていただけなのかもしれない。そんなオレを目覚めさせてくれたこのゲーム、「バンゲリングベイ」。こいつはお薦めだぜ。（支離滅裂御容赦!）

震する戦艦機や爆撃機、高射砲、レーダーなんかはバルカン砲でなぎはらうんです。」  
N「爆弾が9発しかなかったら、途中でなくなっちゃうんじゃないの?」

芋「そんなときのために、海上で味方の空母が巡航しているんですよ。弾がなくなったらそこに着陸すれば、爆弾を補給してくれるし、ヘリコプターのダメージも回復するし、いい。それがなくなったらゲームになりませんよ。」

N「でも、バンゲリング帝国の領海内を、敵の空母が巡航なんかしてたら、袋叩きにあっちゃうんじゃないかい?」

芋「袋叩きっていうのはないですけど、敵の爆撃機や戦艦がやってきて、空母を攻撃したりするってのはあります。だから、工場を爆撃している途中なんかに警報が出たら、空母の救援にいかなきゃならないんですよ。」

N「でもさあ、それはおかしいんじゃないの? だって、普通の空母だったら、防御のための戦力ぐらいあるんじゃない? だいたい、ヘリコプター1機だけで戦争仕掛けるなんて、考えただけでもゾッとしますよ。ヘリコプターしか買えないような国が戦争なんか始めるなっつーの。」

芋「そんなこといったら、ゲームが成り立ちませんよ。」

N「ゲームにしてゲームにあらず。これからのゲームはこれですよ。」

芋「よくワケが分からないけれど、説明を

続けますよ。このバンゲリング帝国には、いくつかの大陸がありまして、その大陸の上には、いくつかの工場があるわけです。そして、プレイヤーの使命っていうのが、この工場を全て、爆弾で破壊することなんです。あつ、これはさっきいつか。だけど、この工場は、爆弾1発じゃあ破壊できなくて、何発かの爆弾を投下しないと、破壊できないんです。しかしながら、1回に積める爆弾の数は9発だから、こまめに補給しなければならぬし、補給にいくと、ダメージを受けた工場は、時間とともに復活しちゃうから、倒しにくくなる。そこで、テクニックが生まれる余地があるわけなんです。」

N「ワープとかはないの?」

芋「あるわけないでしょ。それから、このバンゲリング帝国ってのは、ものすごく広いんです。なんせ、テレビの画面にして100画面分ぐらいでしたっけ? それぐらい広いんですから。そこで今回は、ヒットビットの読者がバンゲリング帝国で迷子にならないように、バンゲリング帝国の島の地図を掲載してあります。参考にしてください。なお、地図を掲載する、というのは、Nさんのアイデアです。読者のみなさんは、Nさんに感謝するように。」

N「私が、ご紹介にあずかりました、オー・ヒットビット編集のNです。この地図を見れば、工場の位置も、敵戦艦の建造ドックの位置も、一目瞭然なんだわさ。」

芋「そういえば、戦艦の建造ドックについていうのを忘れてました。この戦艦というのは、ゲーム開始直後から建造が始まり、完成すると、味方空母を攻撃してくるんです。だから、工場の破壊と並行して、戦艦建造の妨害にもいかなければいけないんです。」

N「レーダーについてもいつてよ。」

芋「あつ、そうそう。そのレーダーっていうのは、攻撃はしてこないんだけど、味方ヘリが近くを通ると、敵戦闘機を呼び寄せたりするという、たいへん陰険なヤツなんです。発見したら、すぐにバルカン砲でやっつけましょう。それから、工場の周りには、たいていレーダーや高射砲が設置されていますんで、工場破壊の前には、必ずそれらを壊しておくこと。これは鉄則です。」

N「さっき芋吉つつあんがプレイしているのを見てたら、破壊したはずの高射砲が、復活して撃ってきた、てのがあったじゃん。あれ汚いよね。」

芋「いや、工場の周辺の地上物っていうのは工場が生産するわけだから、工場さえ健在なら、なんぼでも造ってくるんです。」

N「じゃあ、工場を破壊しなきゃダメじゃん。」

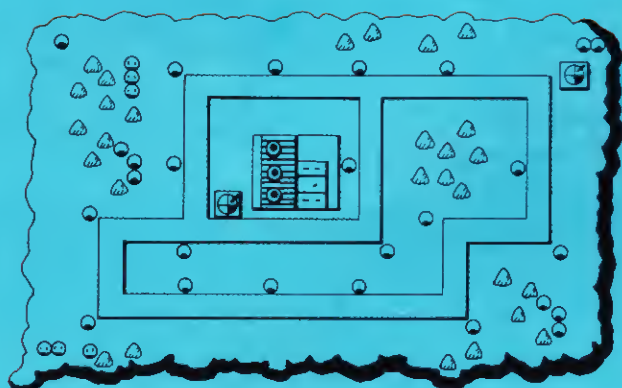
芋「だから、それがこのゲームの目的じゃないんですか。さっきからいつてるでしょう。」

N「考えた者の負けですよ。」

芋吉「……………」



# バンゲリング群島地図



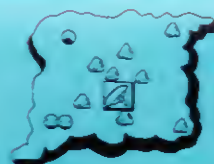
「ヘネモコセ島」 この島は20世紀初頭まで無人島だった。名前の由来は不明。この島には、天然記念物の「ヘネモコセ始祖鳥」が生息している。

「日の丸弁島」 四角い島の真ん中に、赤い工場ボツンとひとつ。いつしか人はその島を、日の丸弁島と呼ぶようになったそう。

「タマウス島」 この島の形が、バンゲリング帝国原産の動物「ヒボポ=タマウス」そっくりなことから付けられた。




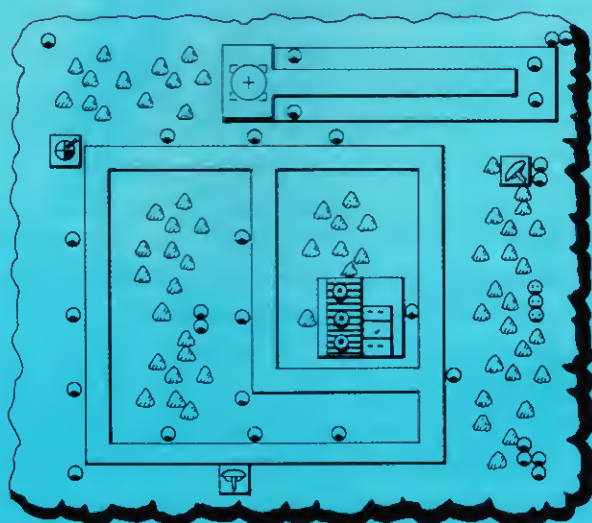
「ジェミニ西島」「ジェミニ東島」 昔この2つの島は1つの大陸だったそうだが、ある夜突然、大地震変動が起こったために、その大陸は2つに引き離されてしまったのだ。ゆえにこの2つの島は、<sup>ジェミニ</sup>双生児と呼ばれている。



 Bungling Factory Locations

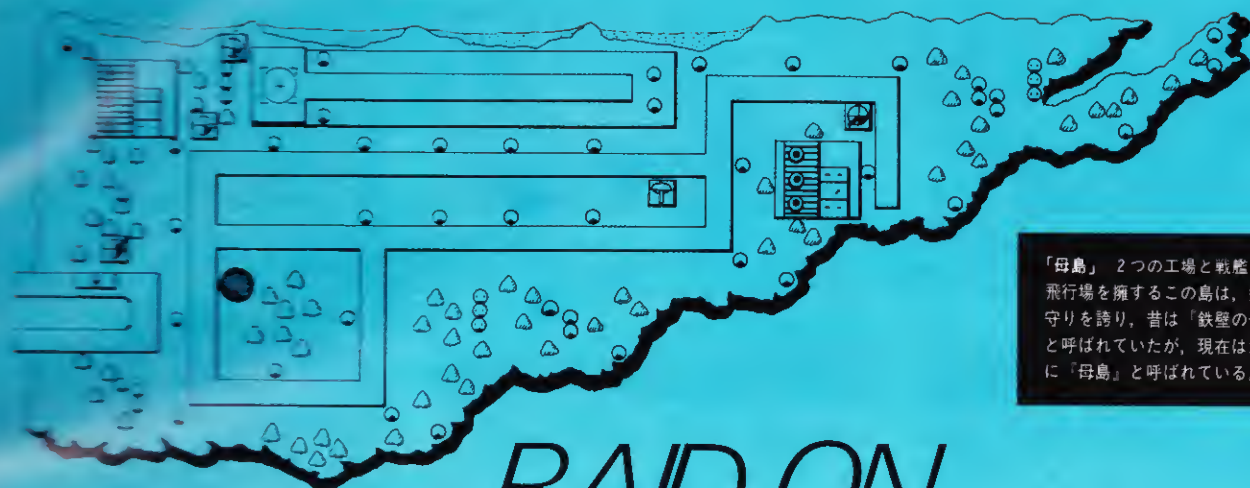
 Bungling Airport Locations

 Bungling Battle Ship Dod Locations



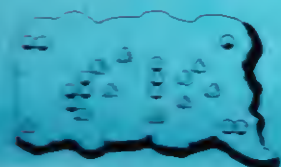
「ファイ島」 この島に空港、じゃなくて飛行場を建設する際に、建設反対闘争があった。その闘争の時の住民のかけ声「ファイト！」がナマって、この名になったという。



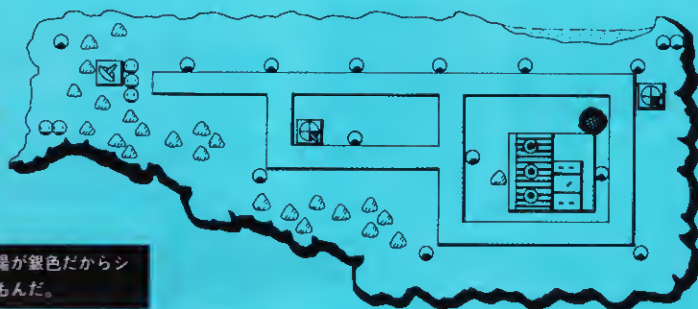


「母島」 2つの工場と戦艦ドック、飛行場を擁するこの島は、鉄壁の守りを誇り、昔は「鉄壁の母島」と呼ばれていたが、現在はただ単に「母島」と呼ばれている。

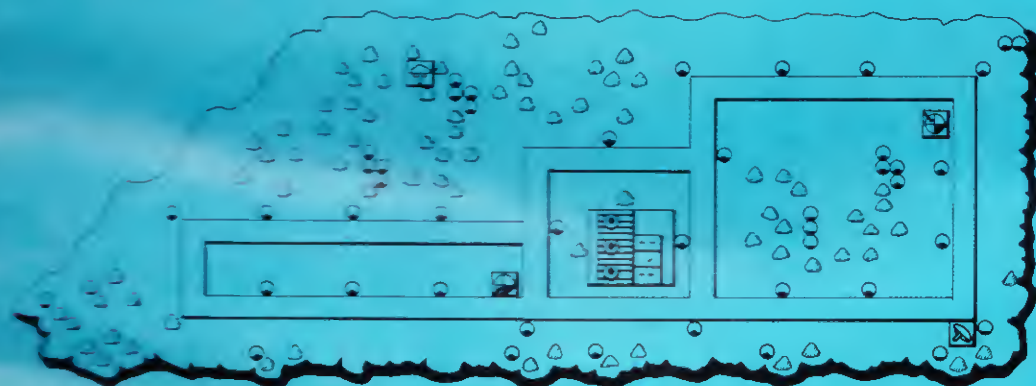
# RAID ON BUNGELING BAY



「シルバ島」 工場が銀色だからシルバ島。単純なもんだ。



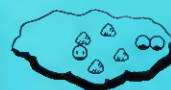
「デモン島」 古代バンゲリング人は、この島を「悪魔の島」と呼んで恐れたようだ。今でもこの島の上空では、原因不明の事故が起こるようだ。くわばら、くわばら。



「メサイア島」 救世主が手を差しのべる形をしたこの島は、きつと天宮にいちばん近い島なんでしょう。



「ダチン島」 「メサイア島」の差しのべた手を、お駄賃をわたす手と見たためについた名







米粒みたいに小さいが、なかなかどーしてあなどれない。坦克の発射するマシンガンは、ヘリに1のダメージを与える。



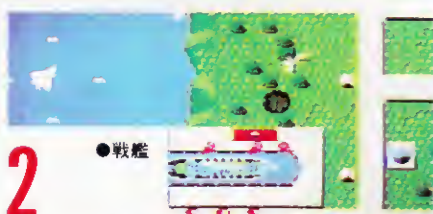
コイツもタンク同様、マシンガンでヘリに1のダメージを与える。たまに炎上しているポートがあるけれど、あれは何なんだろう？



工場を破壊する際の邪魔物。8方向に弾を撃ってくる。バルカン砲で破壊可能。



母島の戦艦ドックで造られる。これが完成してしまったら、空母の命は風前の灯。そうならないように、"CAUTION"のサインが出たら、何をおいても爆撃しよう。



建造途中の戦艦(1)。このままにしておくと、時間とともに戦艦は着実に建造され続け(2)、ついには出港してしまう(3)。こうなる前に爆撃しよう。



キ ャ ラ ク タ



「芋吉の忘れ形見島」この記事の撮影のときに、つい撮り損なってしまっ、後から急に撮り直した島。図にできなくてゴメン。



これがウワサのジェットヘリ。爆弾は9発積載可能。バルカン砲装備。←→のキーで向きを、↑↓のキーでスピードを変えられる。

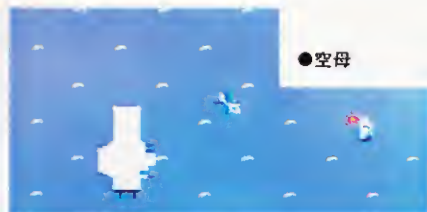
# RAID ON BUNGELING BAY







このシーダーには攻撃能力はないが、ヘリコプターを撃退するのには、戦艦を呼び寄せる、という役割がある。



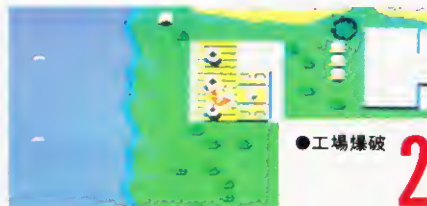
いざというときに頼りになるのがコイツ。ヘリコプターの爆弾を補給し、ダメージを0にしてくれる。着艦・発艦はスペースバーでOK。



この工場を全て破壊するのが、キミの使命。全部で6か所にある。煙を出している間は、高射砲、レーダーなどを生産し続ける。



## 傾向と対策



母島の工場破壊を狙うジェットヘリ。まず周囲の高射砲などを破壊して(1)、工場の真上でホバリングしながら爆撃開始(2)。そして何発かの爆弾を投下すると……みごと工場破壊(3)！



黒いボディの飛行機。味方空母は爆撃するが、ヘリコプターは攻撃しない、という弱い者いじめ(?)の陸軍なセフ。コイツが空母に50発くらい爆撃すると、空母は沈没してしまう。



白いボディの飛行機。ヘリコプターのみを狙ってくるという、嫌なヤツ。バックをとって、バルカン砲で破壊しよう。

### 「バンゲリングベイ」予告編

「バンゲリングベイ」の楽しさは、十分お分かりいただけたと思う。とはいえ発売まであと1か月。待ち遠しいネ。もう待てないキミのために、予告編としてEddy2を使って「バンゲリングベイ」のタイトル画面を作ってみた。詳しくはP.78～P.80参照。



# SONY

楽しみが広がる。  
**ソニーカタログ**

自と映像の新時代  
ベータマックスは、今10周年



QUALITY INNOVATION  
使いやすく、高品質にする。それは、いつもソニーの夢です。

DA PRO 4 HEAD



Hi-Band Beta hi-fi

## パソコン人のビデオ。



HB-701FDでつくったグラフィックやテロップを、ビデオ映像にスーパーインポーズして楽しみたい、さらにこれを編集してテープに収めたい。こんなときに問題になるのが、編集時の画質。そこで、この《ベータプロ》。パソコンと組むのに最適な高画質ビデオだ。

録画画質を飛躍的に高めたハイバンド=Hi-Band採用。ビデオの画質をプロレベルに迫るクオリティに高めたのが、ソニーの新しい映像記録方式(ハイバンド)です。映像記録の帯域を広げ、画像の鮮鋭度を20% (当社比)もアップ。編集を重ねたときの画質劣化が極めて少なく、オリジナルテープが美しく仕上がります。

編集のポイント探しや画像分析に便利なくジョグ/シャトル搭載。放送局のビデオ編集システムの機能を取り入れたくジョグダイヤルとくシャトルリングを搭載。静止画や正逆のコマ送り、スロー、倍速、ピクチャーサーチなどを、お好みの速度で自由自在にコントロールできます。編集時の画像分析やポイント探しが一気に簡単になりました。(写真右:ジョグダイヤル部を手前に開いた状態)



指定区間をワンタッチ編集できる自動編集機能。テープの任意の区間をメモリーさせると、ボタンひとつでその区間を自動再生。これに録画用ビデオ(ソニーの赤外線リモコン対応ビデオ)が連動して自動編集。パソコンと相互コントロール可能なコントロールL双方向端子。将来発売予定のMSXカートリッジを使うと、MSXパソコンと各種の情報を双方向に伝送できる発展端子を搭載。

高音質(ベータハイファイ=Beta hi-fi)。ノイズや歪みを大幅に低減した高純度のサウンドを実現。

さらに充実した新機能 ●映像ノイズを大幅に低減したD.A.プロ4ヘッド採用 ●ひときわ美しい映像記録を可能にした(β1S ベータワン・エス)モード採用 ●編集に最適な画質が得られる(エディットポジション)搭載 ●映像・音声インサート機能 ●コマ送り機能 ●高速アクセス・ピクチャーサーチ ●連続自動頭出し(インデックススキャン) ●自動頭出し(インデックスサーチ) ●音も聴ける倍速再生 ●テレビとFM放送を同時に録画・録音できるサイマルキャスト録音機構 ●FM放送の留守録に便利なタイマー・オンライン機構 ●3選8プログラムタイマー ●クイックタイマー ●2系統AV出力端子など。

# Beta PRO

ハイバンドベータハイファイ SL-HF900 ¥239,800

●多機能ワイヤレスリモコン付属 ●最大外形寸法:幅430×高さ105×奥行397.5mm ●重量:約12kg

**Beta** 録画・再生について ●このビデオはベータ方式のビデオです。βマークのついたビデオカセットテープ以外は使用できません。●β1Sモード(テープ速度:4.00cm/秒)、β1モード(テープ速度:2.00cm/秒)、β0モード(テープ速度:1.33cm/秒)で録画したテープは、それぞれの再生モードのないビデオでは再生できませんのでご注意ください。●このビデオで録画したテープは、ベータ方式の当社及び三洋、東芝、NEC、セナール、アイワ、パイオニア(順不同)各社ビデオのβ1S/β1各モード毎に互換性があります。●このビデオではβ1およびβ1Sモードで録画されたテープの再生ができます。●このビデオはβ1Sモードで録画できます。●外国で作製されたビデオソフトには、テレビ放送方式が異なるために、このビデオデッキでは再生できないものがあります。

●あなたがテレビ放送や録画物などから録画したものは、個人として楽しむなどのほかは、著作権法上、権利者に無断で使用できません。

●カタログ送呈=住所、氏名、年齢、職業、電話番号、機種名を明記の上、〒108東京都高輪局区内ソニー 映像カタログ係へハガキでお申し込みください。

(上手に使用って上手に家電)



# CONTENTS

6	Microcomputer Show'85	
10	MSX新作ゲーム紹介「バンゲリングベイ」	氷水 芋吉
18	読者の広場, 訂正コーナー	
24	ふぞろいのプレゼントたち	
28	ヒットヒットインフォメーション	
33	MSXの通信:RS-232C Interface for MSX	A.Morozumi
40	MSXひょうきんランド	本田武一+大谷英利
46	MSXコンピュータバズラー 第4回	芦ヶ原隆之
51	MSXミュージックエディタII	Jun
56	MSXサウンドエディタ	本田武一
60		
&	メタモルフォーシス	宮入邦男+大谷和利
86		
70	MSX音楽ソフト紹介	栄直
76	MSXミュージックカセット管理プログラム	工藤聖治
78	Eddy2による「バンゲリングベイ・予告編」	
84	ヒットヒットギャラリー	
88		
&	SMCオリジナルゲーム「AREA-X」	中田宏
105		
90	SMCゲーム	
92	リメインズロボット	KAZ
94	F-18誌上コンテスト	
96	SMCグラフィックスエディタを使おう	榎本博行
115	SMCの通信:MIDIインタフェースの制作	西田明宏
120	SMCの通信:SMC用SSTVプログラム	荒井健仁
127	SMCの通信:私の777	普久原朝宏
128	SMCの通信:ほち2号のハード便り	ほち2号
134	SMCリンクパッケージ集 vol.2	
142	見栄3次元CG講座(4)	高柳雄一
146	SMCショートプログラム	落合順子
150	Dr. Logoの世界	友里翔
155	SMC-C講座(4)	乗松保智
160	編集部から	

S U M M E R I 9 8 5

**Vol.6**  
**Oh! HIT BIT**

●CP/M, CP/M-86, Dr. Logoは  
Digital Research社  
●MS-DOS, MSXはMicrosoft社  
の各メーカーの登録商標です。そ  
の他、プログラム名、システム名、  
CPU名は一般に各開発メーカーの  
登録商標です。本文中では、"TM",  
"®"マークを明記していません。  
●本文掲載のプログラムは、個人  
で利用するほかは著作権法上、無  
断複製を禁じられています。



# 読者の広場

DOKUSHA NO HIROBA

## 国内編

▶ Oh! HiTBit も VOL 5 にしてやっとこさ  
まともにお金を出して買ってほしいと思わ  
せる雑誌になった。プログラムを多くのせ  
たからだ。NEC や FM においつけ! おいこ  
せ。 東京都 安藤 滋実 (25)

じゃ、なんです、あなたは今までま  
ともにお金を出してなかった!?

▶ もっと MSX の記事をのせてほしい。SMC  
はもってないので本の半分は全然読まない  
ので 480 円は高くなってしまいます。

埼玉県 延命 淳一 (24)

いい方法があります。あなたが SMC  
を買えば、480 円はとても安いもの  
になります。やったね!!

▶ もっと短いリストをのせて下さい!

愛知県 大天 祥夫 (22)

▶ もっともっともっとショートなショートな  
"MSX" のプログラムをのせてほしい!!

徳島県 佐藤 慶一 (13)

載せてあげましょう。

10 PRINT "Oh! HiTBit" 皆様のこ  
要望にお答えする良心的な編集部なの  
でした。よろしく。

▶ はっきり言って「てか!」と言うのは僕  
はきらいです。てか!

三重県 伊藤 俊史 (17)

▶ テストケイの暗さ、僕にぴったしだっ!  
だからこの本、好きです。

福岡県 平井 正剛 (23)

私も好きですってか。

▶ 「うちのタマ知りませんか」をぜったい  
買います。 三重県 遠藤 信幸 (16)

「うちのタマ…」は良く売れていますね  
エ〜

▶ ゲームプログラムの他に複雑な音楽プロ  
グラムをのせて下さい。(クラシック、映画  
音楽等) 大阪府 安田 進二 (28)

MIDI を SMC に接続する記事があり  
ますので、がんばってみてね。

▶ 編集部の皆様、こんにちは。大へん暑く  
なりましたね。根性でがんばってください。  
ところで今回初めて「Oh! HiTBit」を買  
いました。というのも「SMC-777 C」を買  
もらったところ、この本を見かけ、購入し  
たわけです。内容豊かですっかり HiTBit  
ファンになりました。これからも HiTBit  
ファンの為についてまでもがんばってください。

宮崎県 佐々木 幸枝 (16)

▶ お仕事ご苦労さまで。何だかんだ言  
ってもこの本は貴重な情報源です。ところ  
で、昔むかし NEC の TK-80 という機械は  
ROM 0.75 kB, RAM 0.5 kB, 16 進キーに 8 け  
た LED といういでたちで確か ¥88,  
000 (!) でした。それから約 8 年後が今で  
す。恐ろしいですね。

岩手県 倉田 佳弘 (22)

時の加速度に乗ったマイコン。

マイコンに乗り遅れたワタシ。

▶ おもしろいよお〜 やめられないよお〜  
編集部! どうしてくれる

茨城県 粕谷 良二 (15)

HiTBit やめますか、それとも人間や  
めますか? 私はとくに人間やめて  
ます! \*とね。

▶ やりました。とうとう念願の SMC-777 C  
を父に買わせました。バンザーイ! しかし  
市販ソフトは一本も持っていません。だれ  
かばくにソフトをくれる優しい人はいない

福岡県 宮崎孝 幸三 健悟 (13)

◆Cの勉強がはかばかして来るようになりました。

▶このCの勉強をやる人の過半数は意見した人の中からは、アセンブリが面白い!と思ってはいるけれど、本多君、読本、三浦、2000ユーザの本などは偏見が強い面が強いので、本多君の意見は聞いておいては、アセンブリの勉強がはかばかして来るように思っています。

◆アセンブリの勉強がはかばかして来るように思っています。アセンブリの勉強がはかばかして来るように思っています。アセンブリの勉強がはかばかして来るように思っています。

▶どうもまたアセンブリです。

群馬県 三浦 純裕 (16)

◆アセンブリの勉強がはかばかして来るように思っています。アセンブリの勉強がはかばかして来るように思っています。アセンブリの勉強がはかばかして来るように思っています。

▶このCの勉強がはかばかして来るように思っています。アセンブリの勉強がはかばかして来るように思っています。アセンブリの勉強がはかばかして来るように思っています。

千葉県 三浦 純裕 (16)

▶アセンブリの勉強がはかばかして来るように思っています。アセンブリの勉強がはかばかして来るように思っています。アセンブリの勉強がはかばかして来るように思っています。

福岡県 河野 雅子 (24)

◆アセンブリの勉強がはかばかして来るように思っています。アセンブリの勉強がはかばかして来るように思っています。アセンブリの勉強がはかばかして来るように思っています。

▶アセンブリの勉強がはかばかして来るように思っています。アセンブリの勉強がはかばかして来るように思っています。アセンブリの勉強がはかばかして来るように思っています。

買ったのになア。しかし8bitマシンでUNIXがはしったらキョーフだなア。

鹿児島県 阿久根 大介 (16)

私は4.1BSD版UNIXをいじっていますが、確かに使い易いと思います。ただしUNIXがスペテとは思いませんが。ところで私はUNIXライクなファイルシステムを作った経験があり、CP/M上で走るようにしてあります。これも機会があれば発表したいと思います。

▶C言葉に対して素人の私にとり、乗松さんのSMC-C講座は唯一の光です(もっと紙面を拡げてくれ!)。Sony Filerもいいのですがイマイチ汎用性に欠けます。777(C)のユーザは全員CP/MとCを買い、乗松さんをリーダーにして、マッキントッシュのようなユーザインタフェースを持った統合ソフトの開発に取り組ましよう。

大阪府 富山 重文 (31)

ザマン・リム・マース! (乗松)

▶初めて買いました。友人がHITBITを持っているので、運んでいます。以前3年程

プログラマをやっていたので、もうリストを見るのはうんざりですが、ゲームやパソコンの機能などの記事を見ると、「遊びだけでなく又やってみようかな」という気になります。病気かな?

埼玉県 山本 直彦 (27)

マイコンもソフトなければただのハコ。

遊ぶだけにしてもソフトは必要です。

▶SMCアセンブラソースファイル・コンバータのリストで6A00~6AFFが重複していて、6F00~6FFF番地のリストがありませんでした。別の葉書にも書きましたがそのリスト部分コピーして送りかえしてほしい(3ヵ月も待ちたくないの)。今後、このようなことがない様にしてほしい。

千葉県 内藤 雄二 (20)

▶編集部の人、まじめに働いているんですか。

宮城県 神村 正樹 (11)

なはは、こりやまたシツレイいたしました。編集者はいたってマジメに働いているんですけどね。ところで前回のアセンブラ・ソースファイル・コンバータですが、発見された少々のバグ

愛知県 島 文男 (30)





を修正し、バージョンアップしましたのでその変更部分を掲載しました。どうかゴカンベンチ♡

▶VOL.5 にのっていた「LOSE RUNNER」を入れて、RUNしてみると、金かいを取ろうとしても、すぐにころんで死んでしまいます。穴もほれない。金かいも取れない、面クリもできない。ヒサンやで〜(いれてソシした)。兵庫県 宮城 征人 (14)

それは、さまよえる LOSE RUNNER の魂が、正常なゲームの進行を妨げているのに違いありません。すぐに教会に MSX を持って行って、聖水をかけてもらいましょう。その結果については責任持てませんが……。

▶男はロードランナーに限る

東京都 中村 聡 (19)

ロードランナーは生に限るぜ。

▶なるほど、あの松田聖子の CG はああいう高価なものを使っていたのか。グラフィックエディターではできない訳だ。それから Sony さん、新 Soft 発表発売は嬉しいのですが MSX に将棋を出したら今度は SMC にもお願いします

京都府 木下 峰和 (19)

なるほど、あのグラフィックは CG を松田聖子してる訳だ。それから Sony さんから、「SMC 将棋」6月21日に新発売ですよ。7,000 円。

▶聖子さんが結婚した後、SONY の看板娘は誰になるのですか？やはり「菊池桃子」なのでしょうネ 大阪府 吉川 幸一 (17)

「僕は菊池桃子よりも原田知世がいいんだけど……」「塩沢トキだったりして(笑)」「山口百恵なんか、アナじゃねえか？」「飯島真理を連れてこーい！」「お代官さま、娘だけは勘弁してください」「ならぬ、松田聖子は人妻じゃ」「いえ、まんだでござえます」

▶少しは DISK 関連の記事を出してください。それと、テープ及び DISK でのプログラムサービスをしてもらえればと思います 群馬県 河原 隆 (30)

▶内容がゲーム中心に流れなく非常に良い。今後共よい Tool の提供をお願いします。disk 版も出してはどうですか

宮城県 斎藤 賢吾 (31)

▶Oh!○○や Oh!××には広告の数では負けても中味で勝っているのはよろしゅうおます。この上はスタッフの写真(特に女の方)をさりげなく載せてみては!!

群馬県 市原 誠三 (31)

おっと！まだ、全国指名手配にはなりたくないだよお〜。

▶最近ホストを NEC 製から 777 C に替えました。しっかりとしたマシンのコンセプトに満足しています。1年もたたないうちにモデルチェンジを繰返し、陳腐化による新規需要を作るメーカーの姿勢に怒りを覚えます。これからは、落付いて 777 C と Oh! HiTBit につきあっていけるとおもいます。本誌 VOL.1 が手に入りません。コピーでもいただけないでしょうか

千葉県 佐藤 中一 (36)

Sony はオラたちの神様じゃってか。

▶もうすこし小学生でもわかるような記事を入れてください。

長野県 宮坂 珠実 (11)

わからないことがあったら編集室にドシドシ質問してくらはい。

▶MSX用のグラフィックエディタが欲しいと思っていたところに今回の特集。はつきり言ってくれしかったです。アドベンチャーゲームを作りたいと思っている自分には何かと便利なエディタなのです、ハイ。

京都府 山口 純哉 (18)

投稿が楽しみだな、これは。期待しているよ。

▶年寄りには小さな字のリストが読みにくい!!のです。誌面の都合もあると思いますが、もう少し大きくして欲しい。

東京都 小林 経之 (25)

長時間 CRT とニラメッコをしているわたしは、もう、インクの凹凸だけが頼りです。

▶Vol.1 から Vol.5 まで買いました。あとは本体 (SMC-777) だけなんです。

神奈川県 江沢 隆 (30)

▶59年12月4日 777 C を買い、皆さまの仲間入りさせていただいた高校2年生です。自分でかせいでかったものですからうれしさも倍です。どおぞこれからも長つづきしますように、がんばってちょ！最後にこれから期待してますよ。

三重県 伊藤 和広 (17)

“最後に”だけじゃなくて、最初から期待しててください。マッカセナサ〜イ！

▶強力なグラフィックスエディターがあるのだから、CG コンテストを定期的にやるべきだ。(うちにも秘蔵のピクチャーファイルがあるぞ) 長崎県 池田 文男 (?)

かくさないで、見せてくださいよ。

▶僕はつい最近まで PC-6001 のユーザーだったんですけど、パソコンを買いかえようと思い SHOP に行った。SHOP のおやじは「SMC-777 C はすばらしい」といって私に買わせた。しばらくして、僕は「しまった」と思った。ソフトがすくないのである。しかし、今はすばらしいマシンだと思う。

千葉県 落合 直樹 (14)

どうだ、まいったか。実は、わたしがその SHOP のおじさんだったんですね。

▶まだパソコンを買ったばかりなのでよくわからなかったので本屋へ何か本はないかと買いに行ってみたらくさんパソコンの

お返事いただきました。この「Oh! HITBIT」が  
 読者の目につきました。表の絵がとて  
 めが好い。本が面白かったのです。目  
 づきました。山梨県 大塚 康弘 (14)

お返事ありがとうございました。

▶新製品をいろいろして多1のせてほし  
 三原県 志田 忍二 (13)

▶この雑誌を見て見たのは、たしかVOL.4  
 かなと思います。いいな、と思って  
 書店まで行ってみると、うれ  
 しいです。でもこんどは、手元にお  
 返しをさせていただきます!!

愛知県 阿波井 重幸 (15)

メデタシ、メデタシ。

▶アンケートはがきにミシン目の穴を付け  
 て下さい。毎回切り取るのに苦労しとりま  
 す。今度カミソリの刃を送りましょうか?

神奈川県 臼田 久行 (26)

今どきカミソリはやりませんよ。今  
 はケンシロウを送るのがハヤリです。

ぜひ、ケンシロウを送ってください。

▶最近、読面をMSXにおされぎみのよな  
 気がします。SMC信者は、この雑誌しか、  
 たいした情報やプログラムが得られないの  
 です。777Cが出てから1年たち、影もうす  
 くなり、友だちから「SMCなんか、やめと

け」と言われても、SMCを信仰する信者が  
 いることをお忘れなく。これからもがん  
 ばってください。岐阜県 堀 崇真 (15)

▶"HITBIT INFORMATION"の写真での説明  
 をもっとふやしてより身近なINFORMA-  
 TIONをと願っています。

静岡県 望月 雅貴 (16)

▶書店でOh! HITBITを発見して感激しま  
 した。ほれ込んで買ったハズのSMCだった  
 けど、雑誌にはSMCの記事がまるで載ら  
 ず、内心後悔しはじめたところでした。無  
 人島から生還したような気持ちなのです。

千葉県 田島 康夫 (24)

DOKUSHA NO HIROBA

## 外国編

「海外からの便り」は、遠く地球  
 国エクアドル・キト日本人学校中学  
 生高さんからです。きれいな

切手を貼った封筒で、大事に愛読者  
 カードを送ってくださいました。こ  
 れからもよろしくお願いします。



### 愛読者カード

プレゼントNo.

17

編集部へひと言

数日経てか、HITBITと  
 HITBITと書いてある本が、  
 最近、よく目につく。これは、  
 なんと777Cの雑誌か、  
 エクアドルへ行ってから、  
 いちばん良かった記事  
 として、HITBIT INFORMATION  
 載せてほしい記事・内容

ソフトプログラム (ゲーム)

ソニーのパソコンをお持ちですか?

HB-55 HB-75 HB-101 SMC-777 (SMC-777C) SMC-70

ソニー以外のパソコンをお持ちですか?

機種名:  
 市販のソフトを何本をお持ちですか  
 ( ) 本

パソコン以外の趣味・その他  
 音楽、映画、...



## DEBUG

### Vol.5 SMCアセンブラ・ソースファイル・コンバータ

前号のアセンブラ・ソースファイル・コンバータのリストに一部抜けがありました。また少々バグがありましたので変更点および抜けの部分のリストを訂正リスト1に示します。先月号のリストを打ち込んだ後、リスト1の部分を変更してください。

発見されたバグはインテル型式のニモニックからANNおよびザイログ型式のニモニックに変換する部分でした。また、メッセージの一部も変更となっています。

### Vol.5 Zotgraph

前号Zotgraph記事中にミスプリントがありました。

まず、P51の1行目、16Kシステムに加えるリストは⑦ではなく⑧です。⑦はメインルーチンともいえる部分ですからメモリサイ

ズを問わず打ち込まれていなければなりません。

次に、P52の左下のアミがかかったオマケのリストの説明は、aがディスク用、bが32Kバイト以上のテープ、Cが16Kバイトのテープシステム用のものです。

最後に、図Dのキーボードですが、F4とF9、F5とF10の説明が入れ替わっています。つまり正しくは、F4がセーブ、F9がロード、F5が退避、F10が取り出しです。SELECTキーの下にもF5とF10の説明がありますが、これは削除してください。ESCとTABの説明のところがECCYTABとなっていますが、このYも削除です。ブラシについてはCTRL+Aのみがブラッシングで、CTRL+QおよびCTRL+Zキーがブラシパターンチェンジです。

バグのお問い合わせは、

〒102 東京都千代田区四番町2-1

(株)日本ソフトバンク出版部

Oh! HiTBiT編集係

へ、封書でお願いします



## 訂正リスト

|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum
0E00:	11	9D	0B	CD	99	04	7D	B4	C2	28	0E	2A	79	04	EB	21	:FF
0E10:	99	0B	CD	DB	65	3C	C2	1F	0E	21	1D	0A	CD	FB	09	2A	:1F
0E20:	87	04	22	85	04	C3	56	0E	CD	38	75	2C	02	4D	44	2A	:C0
0E30:	7B	04	EB	2A	87	04	CD	59	0B	22	85	04	C3	56	0E	7D	:9F
0E40:	B4	0A	56	0E	CD	38	75	2C	02	E5	59	50	2A	87	04	C1	:8E
0E50:	CD	59	0E	22	85	04	2A	79	04	5D	54	3E	09	E5	CD	19	:46
0E60:	66	3C	C2	6F	0E	21	1D	0A	CD	FB	09	D1	2A	85	04	E5	:5F
0E70:	CD	DB	65	3C	C2	7D	0E	21	1D	0A	CD	FB	09	E1	CD	B0	:0D
0E80:	73	E5	2A	7F	04	11	0B	00	CD	89	74	29	29	29	01	0B	:6C
0E90:	C0	09	D1	19	22	7F	04	21	00	01	39	CD	B0	73	7D	B4	:14
0EA0:	CA	78	0E	2A	7F	04	EB	21	0B	00	CD	6E	74	D2	E3	0E	:03
0EB0:	EP	11	20	00	CD	6E	74	D2	E3	0E	2A	79	04	EB	3E	09	:67
0EC0:	CD	19	66	3C	C2	CD	0E	21	1D	0A	CD	FB	09	2A	7F	04	:E8
0ED0:	11	0B	00	CD	89	74	29	29	29	01	0B	00	09	22	7F	04	:15
0EE0:	C3	A3	0E	2A	79	04	EB	21	C0	01	39	CD	DE	65	3C	C2	:6C
0EF0:	FB	0E	21	1D	0A	CD	FB	09	2A	79	04	EB	21	A1	0B	CD	:48
Sum :	21	B3	2B	40	EB	F5	B4	92	C0	07	5E	4E	D0	1F	CC	CB	:5E
Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum
0F00:	DB	65	3C	C2	C4	0B	21	1D	0A	CD	FB	09	C3	C4	0B	21	:D9
Sum :	DE	65	3C	C2	C4	0B	21	1D	0A	CD	FB	09	C3	C4	0B	21	:D9
Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum
0F70:	65	3A	09	54	52	41	4E	53	09	73	6F	75	72	63	65	66	:30
0F80:	69	6C	65	20	64	65	73	74	66	69	6C	65	0A	00	00	00	:B4
Sum :	DE	A6	6E	74	B6	A6	C1	C7	6F	DC	DB	DA	7C	63	65	66	:E4
Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum
1260:	91	04	EB	CD	A3	0B	C1	E1	CD	5A	67	2A	91	04	CD	5A	:11
Sum :	91	04	EB	CD	A3	0B	C1	E1	CD	5A	67	2A	91	04	CD	5A	:11
Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum
5110:	18	51	11	BD	50	C3	18	51	11	C7	50	00	2A	A2	4F	7E	:77
5120:	B7	C2	3A	51	23	4E	23	46	C5	D5	CD	14	75	E5	21	03	:D7
5130:	00	CD	C0	61	C1	C1	C1	C3	86	51	7E	FE	03	C2	46	51	:A3
Sum :	CF	EC	0B	6F	34	D2	FF	5A	5C	ED	9B	12	A2	49	B6	D2	:F1
Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum
5400:	4F	4E	09	41	2F	25	D5	73	29	00	4F	55	54	09	2B	25	:44
54F0:	73	29	2C	41	00	7E	D5	CD	A7	73	FE	47	C2	05	55	21	:C7
Sum :	5C	73	35	32	2C	46	FA	40	D6	73	4D	9E	16	0E	7D	46	:0B
Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum
6F00:	EB	22	D4	6E	11	D4	6E	21	D2	6E	0E	0B	CD	8B	6F	3C	:1C
6F10:	22	18	6F	3E	FF	C9	3A	D2	6E	7E	FE	2E	C2	35	6F	23	:EA
6F20:	22	D2	6E	11	D4	6E	21	D2	6E	0E	03	CD	8B	6F	3C	C2	:ED
6F30:	47	6F	7E	FF	C9	3A	20	2A	D4	6E	77	23	3C	20	77	23	:18
6F40:	7F	20	77	23	22	D4	6E	0E	00	1E	0B	1C	1D	CA	62	6F	:67
6F50:	2A	D4	6C	2F	22	D4	6E	7E	FE	7F	C2	5E	6F	0C	1D	C2	:31
6F60:	4B	6F	2A	D2	6C	7E	B7	04	70	6F	21	FF	FF	C3	7E	6F	:D1
6F70:	0C	00	CA	7B	6F	21	FE	FF	C3	7E	6F	21	00	00	7D	C9	:32
6F80:	42	09	26	25	73	29	20	41	4E	44	23	75	32	84	6F	EB	:D6
6F90:	01	82	6F	EB	32	80	6F	70	27	56	6F	22	86	6F	EB	7E	:05
6FA0:	22	66	6F	22	8B	6F	2E	20	12	B0	6F	AF	32	E5	6F	U5	:34
6FB0:	CA	6A	6F	7E	CD	27	00	21	E7	00	00	6F	7A	85	6F	B9	:8B
6FC0:	D4	C6	6F	2E	FF	C9	3A	6B	6E	7E	C2	22	86	6F	C5	LL	:7E
6FD0:	67	73	2A	8B	6F	77	C1	FE	3A	C2	E4	6F	7D	7F	32	8A	:E5
6FE0:	01	C3	F2	6F	23	2C	06	6F	3A	65	7A	3C	32	85	6F	C7	:22
6FF0:	AE	6F	7A	U5	6F	7C	72	6F	7A	7A	24	3F	BD	52	CF	A6	:
Sum :	2B	1A	02	11	22	3C	1F	5C	8C	77	0A	0F	D5	1B	BE	7B	:38



# ふぞろいのプレゼントたち

今年の夏は暑くなるらしい。こんなときにやれ空や山だと出て歩くヤツは、イモ。夏休みじゅう、ずーっと家に閉じこもり、クーラーのきいた部屋でパソコンゲームをやろうじゃないの。サーフィンかなんかでバンバンに肌を焼き、「ドカタ、ドジン」と呼ばれながら、皮膚ガンで死んでいくよりも、ナゾの色白コゾウのままで「タマサブロー」なんざ、かけ声がかかる方が、はつきり言って、シブイ。わかる!?

とゆーわけで、「ふぞろいのプレゼントたち」にふるってご応募ください。とじ込みの愛読者カードに該当項目をすべてご記入の上、ご希望のプレゼント番号を指定のスペースにひとつ記入して、お申し込みください。締め切りは7月18日到着分まで。石原真理子の悪口を書いたハガキは選外になります。なお、当選者の発表は商品の発送をもってかえさせていただきます。



①



②

③ 提供：ソニーテクノロニクス



④

提供：ソニーエハレディ



⑤



⑥



ニークリエイティブプロダクツ

- |                      |     |                         |     |
|----------------------|-----|-------------------------|-----|
| ①AMポータブルラジオ“ジャンボトロン” | 5名  | ⑦キャラクターグッズ「うちのタマ知りませんか」 | 4名  |
| ②デジタルクロック            | 5名  | ⑧キャラクターグッズ「ポチもさがしてください」 | 4名  |
| ③セイコーオーツアラムクロック世界時計  | 5名  | ⑨キャラクターグッズ「ベンジャミン」      | 4名  |
| ④防災用常備電池             | 10名 | ⑩マリーンスカウト、バイクン君キーホルダー   | 20名 |
| （フラッシュライト1本、単1～単3電池） | 10名 | ⑪ミニキンチャク（3枚セット）         | 10名 |
| ⑤バッテリーチェッカーBC-551    | 10名 | ⑫HiT BiT word Tシャツ      | 10名 |
| ⑥セカンド電池              | 10名 | ⑬バッジ & ステッカー（5点セット）     | 28名 |



# ふざけるいのプレゼントたち II





19

SONY

MICRO COMPUTER GAME DISK SMW-G709D  
SMC Floppydisk

21

SONY

SMC Floppydisk

法隆寺の謎



22

SONY

MICRO COMPUTER GAME DISK SMW-G720D  
SMC Floppydisk

23

SONY

SMC Floppydisk

SMC将棋



SONY

MICRO COMPUTER GAME DISK SMW-G716D  
SMC Micro Floppydisk

20

- ⑭ スーパーサッカー(MSX)
- ⑮ α・スクアドロン(MSX)
- ⑯ サイオン(MSX)
- ⑰ MSX将棋
- ⑱ バンゲリングベイ(MSX)

5名

5名

5名

5名

5名

- ⑲ スターブレイザー(SMC)

5名

- ⑳ アズテック(SMC)

5名

- ㉑ 法隆寺の謎(SMC)

5名

- ㉒ リメインズロボット(SMC)

5名

- ㉓ SMC将棋

5名



# 次々増えるヒットビットの仲間たち

## HITBIT HB-201 & HITBIT WORD

去る5月22日から25日の4日間、例年のように東京平和島の東京流通センターで開催された“マイクロコンピュータショウ'85”に出品され、新たにヒットビットファミリーの仲間になった2製品は、オーディオ、テレビ、ビデオなど、ソニーが従来から追求しているファッショナブルな新感覚あふれるデザインをもつものだった。流麗なフォルムのMEZZOのRAM64Kバイト版“HB-201”と事務機器のイメージを払拭し、いつでも、どこでも使える3電源方式の日本語パーソナルワードプロセッサ“HW-30”だ。



### RAM容量64KバイトHB-201

ソニーは、ファッショナブルデザインの新感覚パソコン HITBIT HB-101 をベースにして、より機能を充実させ、64Kバイトの大容量RAMを実装したMSX規格のパーソナルコンピュータ「HITBIT」HB-201(59,800円)を6月1日から発売している。昨年ソニーは、いままでのパソコンの概念を破る、誰でも手軽に楽しめるMAX仕様のパーソナルコンピュータHB-101を発売、マニアはもちろん、パソコンに関心のなかった人々からも好評も博した。このHB-101の上位バージョンであるHB-201は、RAM容量を64Kバイトに増大させ、教育用、生活設計、財務管理など暮らしに役立つソフトウェアにも対応できるコストパフォーマンス機である。

#### 特徴

①RAM容量64Kバイト：現在市販されているMSXソフトはすべて使えるほか、複雑なプログラム作成も可能となる。さらに、フ

ロッピディスクドライブ(HBD-50 89,800円)など周辺機器への接続も簡単に行えます。

②斬新なデザイン：使いやすさを追求した新しいデザインのパーソナルコンピュータです。本格的フルストロークキーボード、片手で簡単に運べるキャリッジハンドル、また、電源コードは本体に収納できるようになっているなど機能性や操作性を考えた数々の機能がコンパクトに収められている。カラーは高級感あふれるツヤ消し仕上げの黒、ライトグレーの2色。

③「HITBITノート」機能：本体に住所録、スケジュール、メモの3種類のデータが処理できるソフトウェアを内蔵。作成したデータは別売のデータカートリッジ「HBI-55」(9,800円)を本体のスロットに差し込むだけで簡単に書き込み、読み出しができ、データカートリッジにいっぱいとなったデータはカセットテープおよび3.5インチマイクロフロッピディスクに簡単にコピーできる。作成した住所録などのデータはアイウエオ順や番号順に並べかえることや、年齢や血液型などをキーワードにして自由に検索す

ることができる。

④ポーズボタンとカーソルジョイスティック：ゲームの途中で、急な用事や電話がかかってきたときに、ゲームを一時中断することのできるポーズボタンが付いている。また、カーソルキーは付属のスティックを差し込むことにより、使いやすく便利なカーソルジョイスティックになる。

⑤グラフィックスとサウンド：256×192ドットで16色のカラー表示ができます。さらに32枚のスプライト画面を使えば、アニメーション作りも楽しめる。また、低音から高音までオクターブの音域と三重和音により、効果音やパソコンミュージックの制作や演奏も楽しむことができる。

#### 仕様

CPU：Z-80A相当  
メモリ：ROM…32Kバイト(MSX BASIC)  
+16Kバイト  
(内蔵ソフトウェア)  
RAM…メイン：64Kバイト  
ビデオ：16Kバイト

画面表示 テキスト表示能力…40w×24l  
 グラフィック機能…256×192dot  
 スプライト画面…32枚  
 カラー表示…16色  
 サウンド機能：8 オクターブ  
 3 重和音+1 効果音  
 カセットインタフェース：1200bps/2400bps  
 FSK方式  
 CRTインタフェース：RF出力、コンポジットビデオ出力  
 キーボード：フルストロークキーボード  
 ASCII 配列および50音配列  
 プリンタインタフェース：セントロニクス  
 社仕様準拠、8  
 ビットパラレル  
 その他：MSXカートリッジスロット2基  
 ジョイスティック端子2個  
 オーディオ出力端子  
 ポーズスイッチ付  
 カーソルジョイスティック  
 キャリングハンドル（収納式）

電源/消費電力：100V 50Hz/60Hz/26W  
 大きさ：380(幅)×275(奥行)×65(高さ)mm  
 重さ：3.1kg

付属品：RFケーブル、アンテナ切換スイッチ、カセット接続ケーブル、カーソルジョイスティック用スティック、BASIC入門、HITBIT ノート機能解説書、取扱説明書

## 簡単ワープロHITBIT WORD

ソニーは、初めての人でも簡単に使いやすく、どこへでも自由に持ち運べるハンディタイプの日本語パーソナルワードプロセッサ「HITBIT WORD」HW-30<sup>o</sup> (98,000円)を6月21日から発売する。

日本語ワードプロセッサは、オフィス用文書作成機として、近年大変普及してきており、昨年度は21万台、85年には45万台以上の出荷が見込まれている。

ソニーでは、1981年からアメリカ向けにオフィス用の英文ワープロ「シリーズ35」を発売し（国内向けは1983年販売開始）、使いやすさと豊富な編集機能で高い評価を受けていたが、この英文ワープロ開発で培った

設計技術やノウハウを活かし、この「ヒットビット ワード」が商品化されたわけだ。「ヒットビット ワード」は、薄型（64ミリ）、軽量（約2.6kg本体のみ）で、従来の事務機のイメージを破るシンプルなデザイン。持ち運びに便利なキャリングハンドル付きだ。また、乾電池、充電式電池、AC100V（電灯線）のどれでも作動する3電源方式を採用しているの、オフィス、家庭、学校から街の喫茶店、列車のなかなど、いつでもどこでも気軽に使える個人用の文書作成機といえよう。

## 特徴

- ①50音順配列キーボード：「あいうえお」をそのままキーボードに配列してあるので、ワープロが初めてという人でも直ぐ打てる。キーボードは、大型のキーを採用したフルストロークタイプなので、操作しやすく、長時間使用しても疲れにくい設計になっている。
- ②かな/ローマ字入力による漢字変換方式：かな文字、英文字のどちらで入力しても漢字に変換できる、かな/ローマ字入力漢字変換方式を採用。英文字は、英文タイプライタと同じ配列（JIS規格）になっているので、英文タイプライタに慣れている人なら、かな文字入力に比べ使用キーも少なく、よりスピーディに入力できる。
- ③3,526文字を内蔵：英文、数字、ひらがな、カタカナ、記号および漢字が使用できる。JIS第1水準の漢字2,965文字を含め、全部で3,526文字を内蔵している。印字の大きさは、全角・倍角・半角（英数字のみ）が選

べ、文章の本文と見出しなどを違った大きさで打てる。

④メモリ機能：よく使う文字や熟語は、常用句メモリで10種類記憶でき、使いたいとき簡単に呼び出せる。また、文書メモリを使えば、400字詰原稿用紙3枚分（1,280文字）まで記憶可能なので、あいさつ状や、案内状、書式の決まっている商用文など、使用頻度の高い文章をメモリすることもできる。スイッチを切ってもメモリが消えないメモリバックアップ機能も付いている。

⑤液晶ディスプレイ：12桁表示のワイドな液晶ディスプレイを採用しているので、打った文章を確認できる。

⑥熱転写プリンタ内蔵：本体に印字音の静かな熱転写式の16×16ドットプリンタを内蔵させている。プリントアウトは、1行ごとに印刷する逐次印刷だけでなく、文書メモリに入っている文章を一括して印刷する全文印刷モードもあり、これを利用して何枚も同じ文書を印刷できる。A4判サイズまでの紙ならば、普通の紙にも印字可能。また、官製はがきや洋封筒など、用途に応じて使い分けられる横書き印刷、縦書き印刷もできる。

⑦カラー印刷機能：印字リボンは、黒、赤、青、緑、茶の5色（別売のリボンカートリッジ「HWR-5C」2,700円）が用意されているので、リボンを交換すればカラー印刷も可。

⑧スマートなデザイン：重さ約2.6kg（本体のみ）、薄さ64ミリと軽量でコンパクト、どこへでも持ち運びができるキャリングハンドルが付いている。ボディカラーは、ライトグレーとレッドの2色。



パーソナル・ワードプロセッサ HITBIT WORD HW-30<sup>o</sup>



# アタマもカラダもアツクなれ!

## BACKGAMMON MSX新作ゲーム SUPER SOCCER

BACKGAMMON 4,000円  
バックギャモン 7/21発売



「ほくドラえもん」

「ほくギャえもん。ドラえもんの兄です」

「あっお兄さん、うしろからクルマがっ!!  
うしろ! バック、ギャえもん。バックギ  
ャえもん。バックギャモン…」

とゆーわけでソニーからMSX用「バックギ  
ャモン」(4,000円)が7月21日発売になりま  
す。1人でも2人でもプレイできるし、ル  
ールをまだ知らないアナタのためには「誰  
れでも解るバックギャモンゲーム入門書」  
が付いてくる。このソフトでバックギャモ  
ンの腕を上げれば、キミも明日から「ギャ  
えもん」。(N)

私が初めてバック・ギャモンと出会ったの  
は、忘れもしないあのひたいから汗の吹き  
出すような、真秋の陽光がキラキラと輝く、  
小春びよりの午後だった。

「ポン」

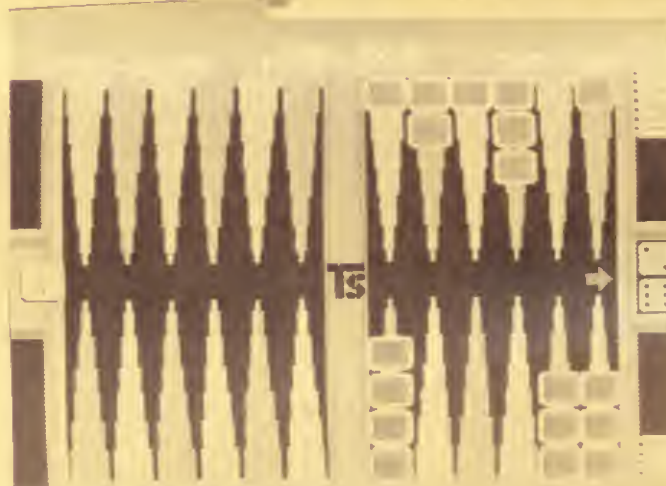
バック・ギャモンの本質とは何だろうか。  
唯2個のサイコロに我と我が人生をたくす  
ことだろうか。相手をヒットすることだけ  
に全精力を投入するなんて、猿にでもでき  
るじゃないか。

「あちよつとまって、それ」。

クローズ・アウトだけを狙って動かす奴が  
いるけど、あれは粘着質だよな。そうじゃ  
ないか。バック・ギャモンてのはスゴロク  
の原型みたいなもんだぜ。要はコマを早く  
上げりゃいいんだろ。簡単なこっちゃない  
か。

「おい、そいつを横するのかよ」

やっぱり、人生の縮図ってところかなあ。  
サイコロの目さえよけりゃ、勝てるもんな。  
6ゾロをそうだなあ、10回もやれば、どん  
な、スーパー達人だって相手じゃねえぜ。  
人間ってのは運だよ運、さもないや、イカ  
サマかな。



「早いリーチは、イー・スー索ノ」

ビギナーズ・タワーって知ってるかい。同  
じ場所に6つも7つも、置ききれない程つ  
み上げる奴がいるんだよ。上手な奴は蟻の  
行進みたいにあつあつ並べているから、そ  
んなことする奴は初心者なんだよ。だから  
さ、上手な奴ってのは長考するんだよ。ある  
程度はめるせるけど、5分も6分も考えら

れると、たまらないよ」

「おい、お前の番だぞ」

ギャモン勝ちや、バック・ギャモン勝ちし  
たことあつか。俺あるぜ。ギャモン勝ちな  
んかよ、誰でもできるけど、バック・ギャ  
モン勝ちはなかなかできないぜ。あれは気  
分いいぜ、最高だぜ。胸がすーとするぜ。

「ロヘン、立直一発面清ツモドラ2!」E

## SUPER SOCCER

4,900円

スーパーサッカー

1/2発売

最近の世は忙ししていると、たまには健康  
なゲームでもしたいなあ、なんて考え  
てみた。ところが、Oh! HiBiTの仕事  
に集中すると、頭脳労働=肉体労働なわけ  
だから、体はヘトヘト、頭はクラクラ。そ  
うして、頭はサッカーのことでい  
っぱいになってしまう、というわけなんだ。  
サッカー サッカー サッカー サッカー  
サッカー サッカー サッカー サッカー  
こんな自分を、私はカワイイと思う……  
ななな

そこで「スーパーサッカー」。これなら、疲  
れた体にムチ打つわけでもないし、動かす  
ものといったら指だけ。さらには、ゲーム  
の操作も超簡単で、僕なんか3回  
目までなのにもう6対0の完封勝利なわけ

だからね。

といって、ゲームの内容が単純なわけじゃ  
あない。むしろ本物ソックリの高度なプレ  
イなんかもできちゃったりするわけで、言  
うなれば、無意識の天才プレーヤー。ほら、  
今だって原稿書きしながらもヘディングシ  
ュートを決めちゃうわけだ。

もちろん、使えるテクニックはヘディング  
だけじゃない。ドリブル、シュートは当たり  
前。ロング・シュートのボールパス、スラ  
イディングタックルに、キーパー操作、コ  
ーナーキックからの攻撃までもができると

いうわけ。それもジョイスティックと2つ  
のボタンの操作だけで。

また、コンピュータとの不毛な対戦に疲れ  
たら、人間相手の2人プレイもできたりす  
るわけだ。人間同士のかけひきで、ポケた  
頭をリフレッシュ!

惜しむらくは、MSX スプライト機能の解  
像度の低さ。でも、それさえ目をつぶれば、  
横スクロールも小気味いいし、審判の動き  
もそれっぽい。

こりゃあ、必見、いや必遊の1本だ、とい  
うわけだー!! (芋)



## よくわかるMSX

入門編・チャレンジ編

各680円



発行・旺文社





## SMC-777/C用 拡張ケーブル

SMC-777/Cで漢字ROMとRS-232Cを両方つなげたい欲張りなユーザーに朗報!!  
SMC-70の豊富な周辺機器(キャッシュディスク8インチコントロールユニットなど)を漢字ROMと同時に使うためのケーブル

(777拡張ケーブル:SCS-777)が株マイコンランド浦和(tel0488-22-3791)から、7月中に発売される。ユーカラJJで通信をしたいとゆうアナタもオメデタイこのケーブル、お値段は2,800円。

# 応答せよ, 応答せよ, 全国のSMCピープル

## ユーザーズクラブ紹介

### SMC SAPクラブ

- ①久道健三
- ②〒150 東京都渋谷区南平台町16-29  
石川ビル4F 練フレックス
- ③男30弱女2。大学生と30代以上が半々ずつ。
- ④なし。
- ⑤SMC-NETが会報。
- ⑥BBS Mailという通信データベースが活動の中心。異常に高度な技術を有するクラブ。
- ⑦とにかくBBS Mail。
- ⑧マニアが多いクラブ。

### SMC(そんでもってクラブ)

- ①金木幸生
- ②〒180 東京都武蔵野市緑町2-6-19-13
- ③男10。大学生から60代まで。20・30代中心
- ④なし。
- ⑤準備中。
- ⑥月1回。銀座 Sony ビルに集合。
- ⑦プログラムの紹介、雑誌のプログラムを打ち込み、デバッグなどを行っている。
- ⑧アマチュアらしい人が多い。
- ⑨マニアは少なく、参加しやすい。

### 名称決定しておらず

- ①鈴木吉明
- ②食住義 〒359 所沢市緑町1-2-1 所沢ビルコ&レッツ3F コレクションステーション
- ③男10。専業主婦と20代くらいまで。
- ④なし。
- ⑤準備中。
- ⑥連絡先になっている店で月1回。
- ⑦雑誌のゲームプログラムを使ってプログラミングの勉強。
- ⑧できたてのクラブ。入るなら今がチャンス。

### 熊谷SMCユーザークラブ

- ①松本 武
- ②〒360 熊谷市弥生町2-47 マイコンショップカーク
- ③男15。17歳から30代前半まで。
- ④なし。
- ⑤準備中。
- ⑥月1回。関東ソニー熊谷営業所で。
- ⑦使うこと中心のクラブ。情報交換、新作ソフト紹介、ソフト交換などを行っている。
- ⑧付近のユーザは気楽に参加してほしい。

### SMCオーナーズクラブ3.5

- ①坂本 徹
- ②〒370 高崎市上中居町331-1
- ③男20。30代以上が半数。
- ④入会金500円、通信連絡費として。
- ⑤準備中。
- ⑥Sony 高崎営業所。2ヵ月に1-2回。
- ⑦集会所が活動の中心。自作ソフトの発表、勉強会など多彩な活動。
- ⑧1人が始めるとみんながバックアップ。
- ⑨息長く続けたい。

### HIT BITマイコンユーザーズクラブ

- ①続木勝年
- ②田中国男 〒462 名古屋市北区上飯田南町4-1 上飯田南荘7棟1001号
- ③男20。学生3名の他社会人。
- ④決まっていない。
- ⑤準備中。
- ⑥月1回最終日曜日。名古屋ソニービルマイコンルームにて、が、まだ正式決定したわけではない。
- ⑦ホビー中心で、ソフトの移植、SMCのBASIC用バックファイルの自作、700のROM改造。
- ⑧オリジナルソフト多数。
- ⑨発展途中のクラブ。

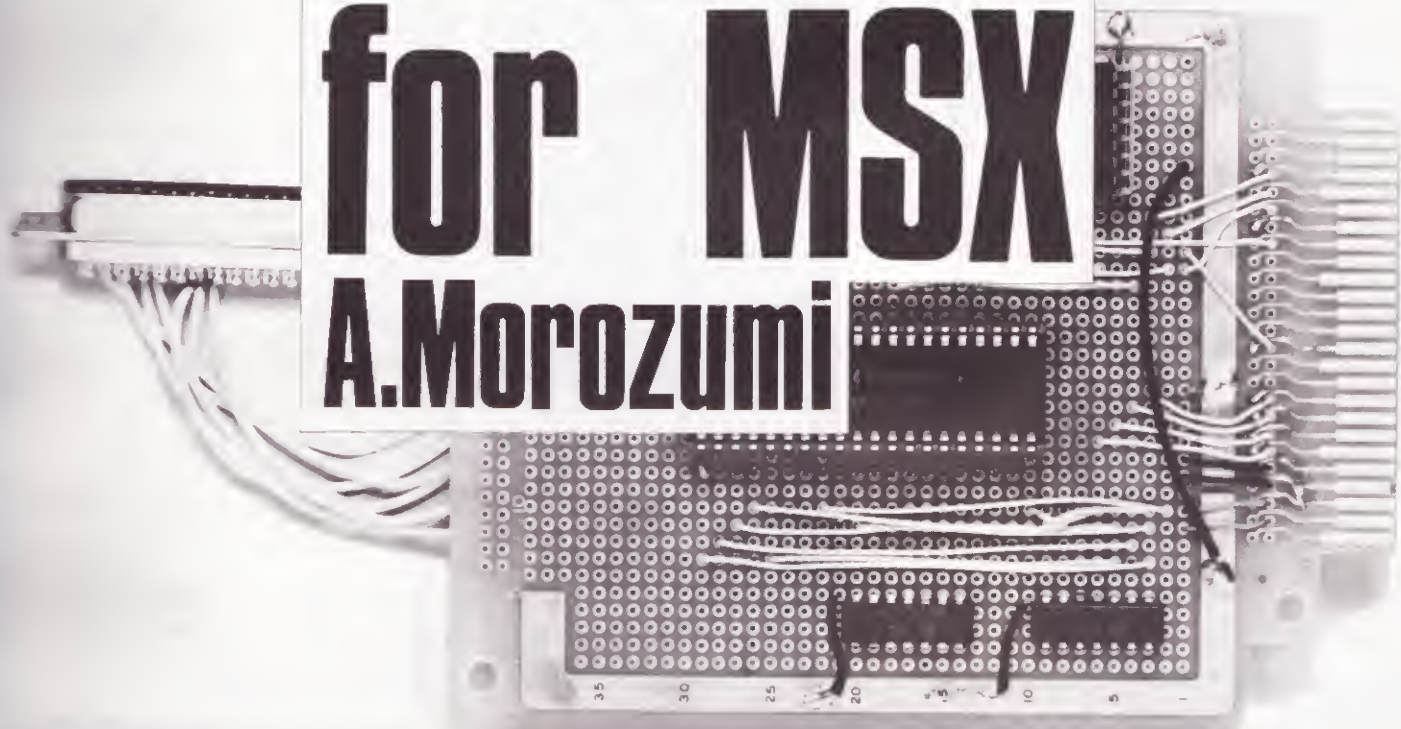
### SMCF(サディスティック・マイコン・クレージー・ファミリー)

- ①小林一彦
- ②〒545 大阪市阿倍野区阪南町6-4-23 ヴィラ大野516号
- ③男30。20代から40代の全員社会人
- ④入会金1500円年会費2000円。今後は未定。
- ⑤3.5インチディスクの供給を準備中。
- ⑥会員が全国に広がっているため。なし
- ⑦⑥のため、水動の中心は3.5インチディスクによる交流。
- ⑧ディスクメディアで交流しているクラブはまだ珍しい。
- ⑨商品化可能なゲームを作りたい。

- 大宮: 〒330 大宮市宮町1-60  
西武百貨店大宮店AVコーナー  
担当: 小増戸  
〒336 浦和市高砂2-8-10  
ナカギンザ共同ビル  
マイコンランド浦和  
担当: 本澤

- 名称 1 代表者 2 連絡先  
3 会員数 4 会費 5 会報  
6 活動場所 7 活動内容  
8 特色 9 メッセージ

# RS-232C Interface for MSX A. Morozumi



## MSXパソコン通信

電電公社の民営化に伴ってテレコミュニケーション、音声カプラ、モデムなどの話題に親しむ機会が多くなってきた。こうなるに、RS-232Cインタフェースの付いているコンピュータはコンピュータじゃないぞという危険な輩が顔を出してくる。電電公社ながらMSXにはRSのインタフェースが標準装備されている機種は数えるほどしかないのが現状だ。これに対す

るメーカーの供給状態はというと、これまたなかなか手に入らないのである。

それでは作ってしまおうと、部屋の隅からほこりにまみれた工具を引っ張り出してきた。「RSインタフェースなんて、モデムやカプラをもっているわけでもないのに、一体何に使うんだ、作るだけ無駄だ」という声もあったが、めげずに黙々と作ることにした。世の中には、無駄と思われるようなことに努力を惜しまない人間がいるが、そこに至るまでの過程を大事にしているかい

ないかによって、一見無駄にみえることも貴重な体験につながるものである。

## インタフェースの設計

何はともあれ、使用するLSIを絞らなければならないので、設計の真似事などしてみようと思う。第一にLSIを決定する必要がある。最近よく使われる行には、インテルの8251や8273、NECの $\mu$ PD 379 USRT、そして今回使うザイログのZ80



図2 MSXスロット

図1 Z80A SIO/Oのピン配列

No.	名称		No.	名称
1	D1	1	40	DO
2	D2	2	39	D2
3	D5	3	38	D4
4	D7	4	37	D6
5	INT	5	36	IORQ
6	IEI	6	35	CE
7	IEO	7	34	B/A
8	M1	8	33	C/D
9	Vcc	9	32	RD
10	W/RDYA	10	31	GND
11	SYNCA	11	30	W/RDYB
12	RxDA	12	29	SYNCB
13	RxCA	13	28	RxDB
14	TxDA	14	27	RxTxCB
25	DTRA	15	26	TxDB
16	RTSA	16	25	DTRB
17	CTSA	17	24	RTSB
18	DCDA	18	23	CTSB
19	CTSA	19	22	DCDB
20	CLOCK	20	21	RESET

Z80A SIO/O

No.	名称	No.	名称
1	CS1	2	CS2
3	CS12	4	SLTSL
5	予備	6	RFSH
7	WAIT	8	INT
9	M1	10	BUSDIR
13	10RQ	12	MERQ
15	WR	14	RD
17	RESET	16	予備
19	A9	18	A15
21	A11	20	A10
23	A7	22	A6
25	A12	24	A8
27	A14	26	A13
29	A1	28	A0
31	A3	30	A2
33	A5	32	A4
35	D1	34	D0
37	D3	36	D2
39	D5	38	D4
41	D7	40	D6
43	GND	42	CLOCK
45	GND	44	SW1
47	+5V	46	SW2
47	+5V	48	+12V
49	SUNDIN	50	-12V

SIO などがある。Z80 SIO に決めた理由は、まず MSX の CPU が Z80 だったことで、本体の CPU と周辺 LSI が同じファミリの方がシステムバスも同じなので信号の変換などがあまりないというメリットがあるからだ。同じ種類のファミリの CPU と周辺 LSI をつなげるのはパズルの要領で、ほとんどの場合には同じ名前の信号同士をつなげばよい。ただし、ファミリの違う LSI に対してはデータバスとアドレスバス以外では、同じ名前でも信号の意味が異なるこ

とがよくあるので注意するべきだろう。メインの石を Z80 SIO に絞ったところで SIO の特徴とそれぞれの信号線を調べてみよう。簡単にいうと、Z80 SIO は同期/非同期の他にも SDLC/HDLC と呼ばれる通信方式にも対応している。入出力ポートがふたつあり、それぞれを独立して動かすことができる。クロック、ピン配列によって 3 タイプ 3 種類のバージョンがある。CPU 同様に Z80 SIO が 250 kHz~2.5 MHz、Z80 A SIO が 250 kHz~4 MHz、Z80 B SIO

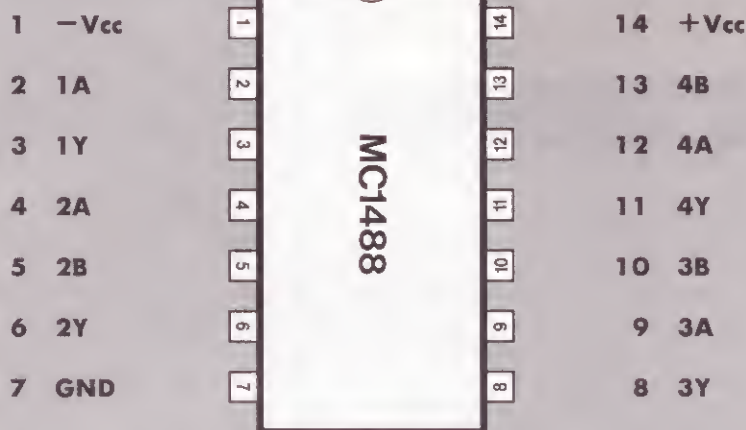
が 250 kHz~6 MHz のクロックに対応している。

SIO の場合は、本来 41 種類の信号があるものを 40 ピンのパッケージに包んであるためにいずれかの信号をひとつにまとめなければならない。そこで、どの信号をまとめであるかによって石の名前の後に 0/1/2 のサフィックスが打ってあるのだ。今回は MSX の 4 MHz のクロックと考え合わせて、Z80 A SIO 0 を使うことにした。LSI の足と信号の関係は図 1 のようになっ

#### 図4 MC1488:MC1489について

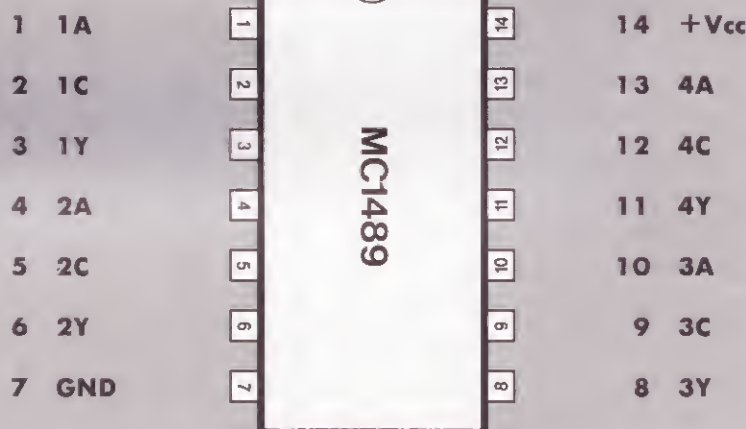
MC1488はモトローラ社のRS-232C用ドライバで、SIOから出て来たTTLレベルの信号をRS-232C用のレベルに変換するものである。+Vccと-Vccには一般的には、+12Vと-12Vの9Vをいれる。しかし、ICの耐圧は15Vまでであるので今回はMSX本体の+12Vを使用する。中身はANDゲートが4個入っているが、実際にはA入力とB入力に同じ信号を入れてNOTゲートとして使う。1のANDだけはIC内部でA、Bの入力をまとめている。入力信号は5VのTTL信号だが、出力は(+、-)のVccによって決まる。入力と出力の関係はVccに(+、-)12Vが、つながっていた場合にA入力とB入力ともに+5Vになったときには出力Vは+12Vになる。それ以外のときには出力Vは-12Vになる。これに対するレシーバであるMC1489は入力Aと出力Bとの間に制御信号のCがある。これは今回は使わない。こちらは+Vcc、5Vで動作するがMC1488とは逆で入力Vは5V以上でも出力YはTTLレベルの5V信号が出て来る。MC1488、MC1489。

#### A MC1488ピン配列



#### 図3

#### B MC1489ピン配列



ている。

この行では27番ピンのRXTXCBが、RXCBとTXCBの合成ということが分かる。これに対するMSXのカートリッジスロットのピン配列を見てみると同じ名前信号がいくつか見つかる(図2)。

以上ふたつの配置の中での共通信号を抜粋すると次のようになる。

MI IORQ RD D0 D1 D2 D3

D4 D5 D6 D7 GND CLOCK RESET  
そしてSIOの入出力信号(RS用)を除くと

残りの信号は

IEI IEO Vcc CE B/A C/D

の6本に絞られる。

この6本の信号の意味を調べると  
IEI, IEO: デイジーチェーン用の割り込み  
イネーブル I/O

Vcc : 電源入力 (+5V)

CE : チップイネーブル入力(アクティブロー)

B/A : チャンネル A/B 切り替え

C/D : コマンド/データ切り替え

となっているので、Vccは+5Vとつないでいいことは分かった。今回はデイジーチェーンは使用しないのでIEI, IEOには結線する必要がなくなる。次にチップイネーブルだが、このRSインタフェースはI/Oアドレス80Hから83Hの間に割り付けられるので図のようになる。

コマンド/データ切り替えをA0、チャンネル切り替えをA1に割り当てるとI/Oアドレスが

80H=A データ



図5 分周回路について

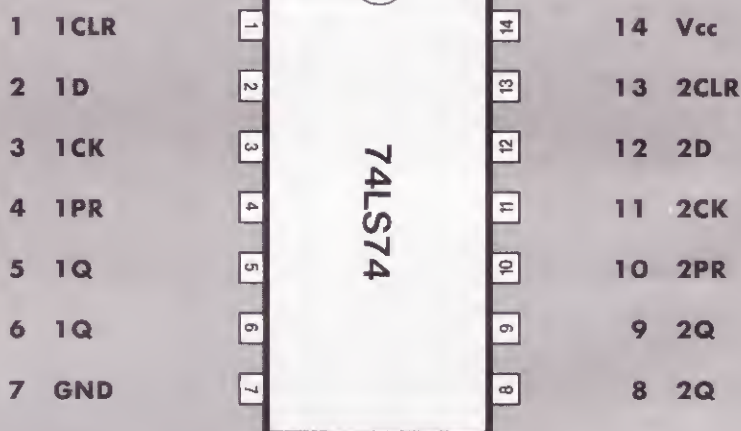
74LS74はD-Flip Flopが2個入ったTTL-ICである。D-Flip FlopはD入力に入った信号をCKの立ち上がりによって出力Qに出す。そこで反転出力Qを入力Dに加えると入力クロックを2分周できるわけだ。

今回はたまたまD-FFを使用しているが同じような分周を行う場合には何もD-FFでなくてもRS-FFやJK-FFなどのFFを利用することもできるのでTTL-ICの規格表で調べてみると面白い。分周回路を含めてクロック関連の部分は、デジタル回路でロジックゲート回路(AND, OR, NOT, NAND, NOR)の次に重要な部分なので、興味があれば調べておく。と後々いろいろなLSIに使えるようになる。同じく分周回路に使われている74LS393はA4入力に入ったクロックがQA出力に2分周、QB出力に4分周、QC出力に8分周、QD出力に16分周されてくる。つまりA入力に16kHzのクロックを入れてやるとQA出力に8kHz、QB出力に4kHz、QC出力に2kHz、QD出力に1kHzのクロックが出てくる。但しVccとGNDにそれぞれ5V、0Vが繋がっていないと動作しないので気を付けなければならない。私はよく『動作しないと大騒ぎしたらGNDが繋がっていなかった』というボカをやるので…念のため。

A 74LS393ピン配列



B 74LS74ピン配列



81 H=A コマンド

82 H=B データ

83 H=B コマンド

に割り当てられる。

以上が、MSX 本体とのインタフェース部である。

これから RS ケーブルへの信号線を作ることになるが、A ポート分のみを作ることにする。Bポート分も作りたいときは A ポートと同じ要領でどうぞ。

ここで使う信号を列挙してみよう。

W/RDYA SYNCA R<sub>x</sub>DA R<sub>x</sub>CA TCA

T<sub>x</sub>DA DTRA RTSA CESA DCDA

W/RDYA: ウェイト/レディステータス出力(アクティブロー)

SYNCA: 外部同期 I/O (アクティブロー)

R<sub>x</sub>DA: 受信クロック入力

R<sub>x</sub>CA: 受信データ入力

T<sub>x</sub>CA: 送信クロック入力

T<sub>x</sub>DA: 送信データ入力

DTRA: データ・ターミナル・レディ出力

RTSA: 送信準備完了出力

CTSA: 送信許可入力

DCDA: データ搬送波検出入力

これらの信号を RS 信号にするにはパワーが弱すぎるので、出力信号にはドライバ、入力信号にはレシーバを入れなければならない。

出力信号は T<sub>x</sub>DA と RTSA と DTRA の 3 本、入力信号は R<sub>x</sub>DA と CTSA と

## 図6 ゲート回路について

ゲート回路についてはご存じの方も多いかと思うが、今回使用した74LS04と74LS10を中心にここで再度認識してほしい。そもそもデジタル回路はTTLレベル(5V=H, 0V=L)の信号によって動作するわけだが、回路的には5Vと0Vの電圧がかかるというより5Vの電圧がかかるか、かからないかということでHレベル、Lレベルが決定される。ゲート回路とはその名のとおりの5Vの電圧をかけるかけないかといことを決める〈門〉にあたる部分である。入力がある一定の条件にあてはまると〈門〉が開いて5Vの電圧が出力に出てくるわけだ。ここでいう条件はゲートの種類によって決まる。例えば74LS10の場合にはA, B, Cの3つの入力に対してすべてに5Vの電圧が生じるという3入力のANDゲートになっている。ANDゲートは入力すべてに対して5Vがかかると出力が5Vになる構成になっている。これに対して、ORゲートはいくつかある入力のうちひとつでも5Vになると出力が5Vになる。そして74LS04に代表されるNOTゲートは入力に対して出力が反転するようになっている。すなわち入力に5Vがかかっている場合には出力が0Vになり、入力に0Vがかかっている場合は出力が5Vになる。真理値表と見比べるとよく分かるだろう。

### A 74LS393ピン配列

1 1A  
2 1B  
3 2A  
4 2B  
5 2C  
6 2Y  
7 GND



14 Vcc  
13 1C  
12 1Y  
11 3C  
10 3B  
9 3A  
8 3Y

### B 74LS74ピン配列ピン

1 1A  
2 1Y  
3 2A  
4 2Y  
5 3A  
6 3Y  
7 GND



14 Vcc  
13 6A  
12 6Y  
11 5A  
10 5Y  
9 4A  
8 4Y

DCDAの3本になる。

買ったのはRXCAとTXCA、そしてSYNCAとW/RDYAである。RXCAとTXCAは送受信のボーレートを決めるためのクロックを入れる。クロックは、よく使われるX16モードだ。これはSIO内部でRXCAとTXCAに入力された信号を16倍にするモードで、SIOの資料ではよく見かける(自分で設計するなんていうのも、せつかくの資料は有意義に使わなくちゃもったいない)。

このモードでは、クロックを13分周すれば19200ボーの転送速度が得られる。しかし、今回はさらに分周して9600, 4800, 2400, 1200, 600, 300ボーまで設定できるようにした。このインタフェースは非同期通信なので、SYNCAはVccにつないでおく。最後にW/READYだが、これはMSXカートリッジにつなぐ。RS用のドライバのVccは+12Vにつなぐよう気を付ける。これでやっと設計が終わった。

ここまでででき上がった回路図(図3)を示す。

やっとハンダごての出番だが、その前に必要な部品は揃っているだろうか。部品リストと照合してみよう。

工具としては、ハンダごて、ニッパ、ラジオペンチ、ハンダ、配線材があれば十分だろう。もしもハンダミスをしたときにハンダ吸い取り器があるとBetter!。配線材は0.3mm位のものを5色ぐらい揃えておくとよい。こては15~20Wのものがほしい、ハ



# RS-232C Interface for MSX

部 品 表

LSI	Z80A-SIO/O	1
IC	74LS74 74LS393 74LS260 74LS04 MC1488 MC1489	2
コネクタ	RS-232C用メス	2
DIPSW	6連以上	1
はんだ	0.3mm前後の太さ 4~5色あった方がいい	1
配線材	60%のヤニ入り	1
基板	今回使用したのは アドテックのMSX用 ユニバーサルボードII を使用(スロット用のパッケージ がついているので使いやすい)	1

## 製作上の注意

今回は、特に難しい点はないので、一般的な注意事項としてまずハンダブリッジといふハンダに注意してほしい。ハンダブリッジとはハンダによるショートのことです。MSX本体に差し込む前にテストで導通をチェックすることをすすめる。テストをもっていない場合は、電池と豆電球とテスト棒で作る。導通のチェックだけなのでこれで十分だろう。もうひとつはハンダはハ

ンダは絶対に60%のを使うべきだ(ハードの製作においては、弘法も筆を大いに選ぶのだ!)ICやLSIにはできる限りソケットを使うべきだろう(たとえ50円のICに160円のソケットを使う羽目になってもその分動作チェック時に非常に楽になる)。一般的にVcc(+5V)には赤、GNDには黒の配線材をよく使う。くれぐれも指をハンダ付けしないように気を付けて…成功を祈る。

ンダがしっかり付いていないことで、配線を引っ張るとすぐにとれてしまうので分かる。

ノイズ対策としてはIC 2、3個にひとつ0.1μFのコンデンサを入れるとよい。入れ方はVccとGNDの間に直接コンデンサの足をハンダ付けする。





MSXひょきんランド

本田武一 + 大谷英利

# A SHORT VACATION

ライナーノーツ

本田 武一

薄く切ったフロップスをアイスティーに浮かべてエ、海に向けたテラスでキーボードを叩く…そんな季節、夏はもうそこである。私が大谷氏に原稿を依頼されたとき、正直いって何を書いてよいものか迷ったものだ。堅苦しいプログラムについての説明は、自

分の物ならまだしも、人の物まではなかなか気疲れするものであるからだ。しかし、

「A SHORT VACATION」の構想を聞かされたとき、私は俄然、乗り気になったのである。日本のポップスの大御所である大瀧詠一氏が（奇しくも氏の所属はCBSソ

ニーである）不朽の名作「A LONG VACATION」を発表してから5年が経過している。しかし今もなお、その輝きを失っていないのは驚くばかりである。大谷氏のいうには、自分のプログラムも何年経っても色褪せないものを造りたいという、またショートプログラムの金字塔を確立したいというコトなのだそうだ（私は、大それた考えであると思うが）。一種の誇大妄想にさえなっているのだろうか。それは、とにかくショートプログラムの「A SHORT VACATION」なんとシャレているではないか。なには、ともあれ楽しい内容である。まずはあなたにをおすすめしてこの稿をしめくりたい。

6月吉日

ライナーノーツのライナーノーツ

H.O タニ

本田君、非解説ありがとう。武一君の話はどーも堅くてイケませんナア。私が、御紹介に預りました教組のH.O タニです。あなたワア、神ヲウ、信じますかア。少しだけエ時間、ありますかア？ とまあ、暇な方

はぜひこのコーナーを読んでいただきたいのです。vol.4, vol.5と連載したひょうきんシリーズも好評のうちに第100回を迎えることになり、今回は記念特集号でありマス(ウソつけ！)しかし、一説によると彼

```
1000 '
1010 ' A SHORT V.A.C.A.T-I-O-N
1020 ' H.O.T
1030 DEF USR1=&H90
1040 KEY(1)ON:KEY(2)ON:KEY(3)ON:KEY(4)ON
1050 GB=0
1060 ON KEY GOSUB 1080,1090,1100,1110
1070 ON GB+1 GOTO 1040,1120,2000,4000,5000
1080 GB=1:DMY=USR1(0):RETURN
1090 GB=2:DMY=USR1(0):RETURN
1100 GB=3:DMY=USR1(0):RETURN
1110 GB=4:DMY=USR1(0):RETURN
1120 'A-1
1130 ' A.S.V.
1140 ' H.O.T
1150 KEY(2)OFF:KEY(3)OFF:KEY(4)OFF
1160 COLOR 15,1,1:SCREEN 2,2:CLEAR
1170 GOSUB 1900
1180 LINE(32,0)-(223,191),15,BF
1190 LINE(56,71)-(199,174),4,BF
1200 RESTORE 1520
1210 READ M$
1220 IF M$="END" THEN 1320
1230 IF M$="I" THEN READ C:COLOR C:GOTO 1210
1240 IF M$="P" THEN GOTO 1830
1250 PSET(VAL("&H"+LEFT$(M$,2)),VAL("&H"+MID$(M$,3,2)))
```

は、僕らとときに1、2の上は「たくさん」  
いってほしい)。

そのほかにもまあ、私としたことが「B.  
C.」をやってミスをしてしまったの  
で、誰か恨んで人を憎まず——筆者は64  
歳まで生きてきたもので16Kではメモリ  
不足により動かないのです。修正  
プログラムが必要です。ゴメンナサイ。

7F040 行	&H AFFF → &H DFFF
7F060 行	&H B000 → &H E000
7F080 行	&H B000 → &H E000
7F060 行	&H B007 → &H E007
7F210 行	&H B0 → &H E0

「早くも気を取り直した強い  
意欲で、新作、『Lose Lunner』について  
ファンと風になんか解説をすることにいた  
る。」

「あれー、おかしいぞー。金塊が取れないぞー。」

「本当だあ。金塊につまづいてるぞ。  
プログラムミスじゃないのかなー。」

「先生！〇タニ先生！」

「さあ、××君、それはねー取れないよう  
 になってるんだよー。そこが、面白  
 いんだよー。分かった？」

三三、三「いいかげんにしなさいッ!!」

―― ③の産図は分かっていただけ





ましたか。よーし日本の将来は明るいぞ！  
 (金塊が取れないと電話くださった方、御心配をおかけしました。)  
 前置き長く、プログラム短く、中身無く H.O タニの趣味のプログラムだけを集めた「A SHORT VACATION」ごゆっくりとお楽しみください。

## A.S.V.

## A-1

えータイトルだけで、ひとつのプログラムという大胆さが受けております。ただデータを取り、絵を描くだけですが、16進データのためメモリをあまり食わないのが特徴です。なんと、お茶にごしの波の音(にはとても聞こえない)が、サービスされております。「A SHORT VACATION」すべてにいえませんが、「A-1」とは、A面1番目のことでファンクションキーにより選ぶことができ、ひとつひとつ個々のプログラムとして分けてもよいようになってます。御覧のタイトル画は、2時間の耐火試験にも合格した安心していただける商品です。お電話のお申し込みは…おーっと口調が Telephone Shopping になっちゃおしまいだよねえさん(なんだ?)。

## ミルミル

## A-2

ドーンと1億健康貯金「ミルミル」の登場です。前号で予告したとおりミルミルプロ



```

2180 '----- BASIC / プリントアウト
2190 BB=PEEK(SB+3)*256+PEEK(SB+2)
2200 L1%=RIGHT$(STR$(BB),LEN(STR$(BB))-1)+""
2210 ST=SB+4 'シマウサノ TEXT / セントウハノチ
2220 '----- メイン222222222222222222222222
2230 IF ST=TB-1 GOTO 2410
2240 S0=PEEK(ST): S1=PEEK(ST+1): S2=PEEK(ST+2): S3=PEEK(ST+3):
      S4=PEEK(ST+4): S5=PEEK(ST+5): S6=PEEK(ST+6): S7=PEEK(ST+7):
      S8=PEEK(ST+8)
2250 IF S0>=H80 AND S0<=HFD THEN GOSUB 2500
2260 IF S0>=H3A AND S1>=HBF AND S2>=HE6 THEN GOSUB 2500
2270 IF S0>=HFF AND S1>=H80 AND S1<=HB1 THEN GOSUB 2500
2280 IF S0>=H22 OR S0>=HBF OR SC=1 OR S0>=H84 THEN GOSUB 2550
2290 IF S0>=H20 AND S0<=H5B THEN GOSUB 2640
2300 IF S0>=H10 AND S0<=H1B THEN GOSUB 2690
2310 IF S0>=HF THEN GOSUB 2730
2320 IF S0>=H1C THEN GOSUB 2770
2330 IF S0>=H1D THEN GOSUB 2820
2340 IF S0>=H1F THEN GOSUB 2900
2350 IF S0>=HD THEN GOSUB 3020
2360 IF S0>=HE THEN GOSUB 3070
2370 IF S0>=H20 THEN L1%=L1%+" ":ST=ST+1
2380 IF S0>=HC THEN GOSUB 3120
2390 IF S0=0 THEN 2410
2400 GOTO 2220
2410 '----- DISPLAY
2420 FOR Z=1 TO LEN(L1%)
2430 Q$=INKEY$
2440 IF Q$="" THEN 2430
2450 PRINT MID$(L1%,Z,1);
2460 NEXT Z
2470 PRINT
2480 L1%="":S8=TB
2490 GOTO 2140
2500 '----- ナイフ コート SUB
2510 IF S0>=H3A AND S1>=HBF AND S2>=HE6 THEN L1%=L1%+" ":ST=ST+3:SC=1:GOTO 2540
2520 IF S0>=H80 AND S0<=HFD THEN L1%=L1%+CC$(S0-128):ST=ST+1:GOTO 2540
2530 IF S0>=HFF AND S1>=H80 AND S1<=HB1 THEN L1%=L1%+CC$(S1-4):ST=ST+2
2540 RETURN
2550 '----- ASCII コート SUB
2560 S0=PEEK(ST)
2570 IF ST=TB-1 THEN 2620
2580 IF S0=34 THEN 2620
2590 L1%=L1%+CHR$(S0)
2600 ST=ST+1
2610 GOTO 2550
2620 SC=0
2630 RETURN
2640 '----- ABC-123 SUB
2650 L1%=L1%+CHR$(S0)
2660 ST=ST+1
2670 RETURN
2680 '----- カス SUB
2690 '----- 0-9 S0>=H11->H1A
      L1%=L1%+RIGHT$(STR$(S0->H11),1)
2700 ST=ST+1
2710 RETURN
2720 RETURN
2730 '----- 10-255 S0>=H0F
      L1%=L1%+RIGHT$(STR$(S1),LEN(STR$(S1))-1)
2740 ST=ST+2
2750 RETURN
2760 RETURN
2770 '----- 256-32767 S0>=H1C
      SS=S2*256+S1
2780 L1%=L1%+RIGHT$(STR$(SS),LEN(STR$(SS))-1)
2790 ST=ST+3
2800 RETURN
2810 RETURN
2820 '----- 32768-999999 S0>=H1D
2830 IF S2>0 THEN F2%=HEX$(S2) ELSE F2%="0"+HEX$(S2)
2840 IF S3>0 THEN F3%=HEX$(S3) ELSE F3%="0"+HEX$(S3)
2850 IF S4>0 THEN F4%=HEX$(S4) ELSE F4%="0"+HEX$(S4)
2860 SS=VAL(LEFT$(F2%,1)+". "+RIGHT$(F2%,1)+F3%+F4%)*(10^(S1-65))
2870 L1%=L1%+RIGHT$(STR$(SS),LEN(STR$(SS))-1)
2880 ST=ST+5
2890 RETURN
2900 '----- 1000000- S0>=H1F
2910 IF S2>0 THEN F2%=HEX$(S2) ELSE F2%="0"+HEX$(S2)
2920 IF S3>0 THEN F3%=HEX$(S3) ELSE F3%="0"+HEX$(S3)
2930 IF S4>0 THEN F4%=HEX$(S4) ELSE F4%="0"+HEX$(S4)
2940 IF S5>0 THEN F5%=HEX$(S5) ELSE F5%="0"+HEX$(S5)
2950 IF S6>0 THEN F6%=HEX$(S6) ELSE F6%="0"+HEX$(S6)
2960 IF S7>0 THEN F7%=HEX$(S7) ELSE F7%="0"+HEX$(S7)
2970 IF S8>0 THEN F8%=HEX$(S8) ELSE F8%="0"+HEX$(S8)
2980 SS=VAL(LEFT$(F2%,1)+". "+RIGHT$(F2%,1)+F3%+F4%+F5%+F6%+F7%+F8%)*(10^(S1-65))
2990 L1%=L1%+RIGHT$(STR$(SS),LEN(STR$(SS))-1)
3000 ST=ST+9
3010 RETURN
3020 '----- GOTO SUB S0>=H0D
3030 B1=PEEK(S2*256+S1+3):B2=PEEK(S2*256+S1+4)
3040 L1%=L1%+RIGHT$(STR$(B2*256+B1),LEN(STR$(B2*256+B1))-1)
3050 ST=ST+3
3060 RETURN
3070 '----- GOSUB&RESTORE S0>=H0E
3080 SS=S2*256+S1
3090 L1%=L1%+RIGHT$(STR$(SS),LEN(STR$(SS))-1)
3100 ST=ST+3
3110 RETURN
3120 '----- &H----- S0>=H0C

```





## A—3

ウーム、英語で題をつけると、何やらかつこいではないか。そもそも「TTF」とは、自画像のことで（さかさに読んでいただく）、パソコンに自分を描いていただくのです。まず拡張スロットに筆を立て、絵の具をキーボード上に出し画用紙はA3版を版画のように押しつけ……るわけがないだろう！ 前衛芸術家じゃないってば。

MSX BASIC には、メモリから V-RAM  
へブロック転送するサブルーチンがありま  
す(ウッ!いきなりマジだ)。そこで、  
BASIC ROM の情報をそのまま送ればよ  
いのです。ここでは、高解像度モードとマ  
ルチカラーモードのふたつにより自画像を見  
ることができます(2 か 3 を入力する)。  
E 011 番地および E 012 番地の内容を変え  
ると(4150 行と 4260 行)、スピードや音に  
変化が出て楽しめると思います。キーボ  
ードから何らかの文字を入力すれば、入力待  
ちの状態になります。成増。そこで、また  
ファンクションキーを押せばよろしい。

## A-4

1985年夏、あなたはとてつもない恐怖を体験する。お手持ちのMSXが人工知能に、それも、夜ごと「シリトリ」で襲いかかってくるのだ。日本語もろくに知らないという君、

[illegible]

## おまけのリスト

```

1090 FOR Z1=1 TO 24
1215 LEVEL=1:NI=5
1240 MX=120:MY=135:P3=4
1270 IF LEVEL=1 THEN GOSUB 3780
1300 IF LEVEL=1 THEN RESTORE 3370 ELSE RESTORE 7000
1425 IF LEVEL=1 THEN M1$="005" ELSE M1$="002"
1430 X1=144:Y1=160:C=8:GOSUB 4430
1445 IF LEVEL=1 THEN M1$="001" ELSE M1$="002"
1450 X1=216:Y1=160:C=8:GOSUB 4430
2060 IF MX=181 THEN ON LEVEL GOTO 2100,5000
2070 IF F=4 AND MX=72 AND LEVEL=1 THEN 2290
2076 IF F=4 AND MX=72 AND LEVEL=2 THEN MX=MX-1
2080 IF F=8 AND MX=170 THEN ON LEVEL GOTO 2540,6000
2970 IF NI=2 THEN LEVEL=2:GOSUB 6900:GOTO 1220
2975 IF NI=0 GOTO 3050
2995 IF LEVEL=2 THEN GOTO 3030
3771 DATA 0,0,60,60,24,88,36,4
3772 DATA 0,0,60,60,24,24,40,72
3773 DATA 0,0,60,60,24,20,18,17
5000 '----- LEVEL-2-1
5010 MX=MX+1
5020 P1=P1+SG
5030 IF P1>3 OR P1<1 THEN P1=2:SG=SG*-1
5040 IF MX=185 THEN PLAY"D4L32CDEFGAB":GOTO 5100
5050 PUT SPRITE P9,(0,209)
5060 PUT SPRITE P0,(MX,MY-2),15
5070 P9=P0
5080 GOTO 5010
5100 FOR Z=141 TO 93 STEP -1
5110 DRAW"BM186,Z;C11R3"
5120 PUT SPRITE P0,(MX,MY-(143-Z))
5140 NEXT Z
5150 SOUND 0,100:SOUND 6,200:SOUND 7,6:SOUND 8,16:SOUND 12,15:SOUND 13,1
5160 FOR ZZ=1 TO 500:NEXT ZZ
5165 SOUND 6,0:SOUND 7,62:SOUND 8,8
5166 PLAY"D4L32BAGFEDC"
5170 LINE(186,93)-(188,141),0,BF
5190 FOR Z=93 TO 143
5200 PUT SPRITE P0,(MX,MY-(143-Z))
5210 NEXT Z
5214 PUT SPRITE P0,(MX,209)
5215 PUT SPRITE 20,(MX,MY),15
5220 GOSUB 2700
5225 T1=152:T2=160:T3=168:E1=180:E2=200
5230 GOTO 2760
6000 '----- LEVEL-2-2
6005 SD=22:SR=1:ST=23
6006 LINE(176,46)-(184,47),0
6007 PLAY"V805L1604CR4ER4GR405L32CR16CR16CR16CR16C"
6010 FOR Z=176 TO 210
6030 SD=SD+SR
6040 IF SD>24 OR SD<22 THEN SD=23:SR=-SR
6050 PUT SPRITE ST,(0,209)
6060 PUT SPRITE SD,(Z,39),11
6070 ST=SD
6080 FOR ZZ=1 TO 25:NEXT ZZ
6100 NEXT Z
6110 PUT SPRITE ST,(0,209)
6120 CLS
6130 FOR Z=1 TO 1000:NEXT Z
6135 PLAY"V805L1604BAGFEDCQ3BAGFEDC"
6140 FOR Z=39 TO 192
6150 PUT SPRITE P0,(170,Z),15
6160 NEXT Z
6170 DRAW"BM170,191C4U20L10U5R10U10R5D10R10D5L10D20L5"
6180 PLAY"S10M4000D4L46L86GL4GLBA+AAGGF+GR4"
6200 GOTO 3050
6900 '-----
6910 FOR Z=1 TO 24
6920 PUT SPRITE Z,(0,209)
6930 NEXT Z
6940 RETURN
7000 ' 1234567890123456789012345678
7010 DATA"
7020 DATA" 4
7030 DATA" 55555555555555553 4
7040 DATA" 4 4 4 4 4 3555553
7050 DATA" 1 3
7060 DATA" 4 13
7070 DATA"32121212121212121212121212 3
7080 DATA"3 555554 3
7090 DATA"3 355553
7100 DATA"3 1 4 4 422 3
7110 DATA"122111322112211 41222113
7120 DATA" 1 1 3 3
7130 DATA" 1 1 3 43
7140 DATA" 1 3 43
7150 DATA" 1 3 4 3
7160 DATA" 3 322
7170 DATA" 3 3
7180 DATA" 3 4 3
7190 DATA"1111111111111111111111111111

```

「ない私達に勝利はあるのか。フッフッフ」 今回のプログラムは、入力している文字（もなかじゃないよ）にもネタが割れないようデータを暗号化してあるのです。これは考えると分かるけど、それは入力してからの楽しみキャラメルということで、おまけすら打ち込んでください。プログラムに注意として「ワ」と「ク」をまちがえないように。RUNしてから始まるまで、ちょっと時間がかかります。ジャンケンは、「か」「ち」「は」で、グーチョキパーです。キーキーと **[CAPS]** キーを押すのも忘れてね。あと、相手の解答の中で分からない言葉が出てきたら辞書を引いて勉強してね。または、アンケートハガキで質問をするという方法もある。その場合よかった記事のところは…

やっそこ、やっそこ、繰り出したてなわけやっそこさ（しつこい）4本立てのように見えます…か？ 1本ずつ入力するには「まま…」マークのGOTO 1040を取り除けば簡単に独立できます。一応B面発表の感想もありますので3か月後に御期待を。ミルミルプログラム作るプログラムを誰か書いてほしい今日この頃です。

## おまけのジミー頁

The Return of The Lose Rumer というわけで前回引退しておきながら今回も登場いたします。今月号、本文にも書きましたが、金銭は取れません。あまりにお問い合わせが多く、えーいそれならこっちにも考えがあるんだぜノっと前回のプログラムの合間と続きに御入力お願いいたします。はっき言って第2作はつまらんですよノ入れても無駄ですよ。絶対入れない方がいいノコラフ。入れるなッノ（編集部に怒られてしまいそーだ）それでは、途中ですが時間でるので野球中継を終わります。



# MSX Computer Puzzler

## 猫 の 夏 休 み

芦ヶ原隆之

第4回 隣の猫はよく団子食う猫だ、の巻

もうすぐ太陽の季節の到来だね。天気の良い日には外に出て、思いっきり遊ぶと最高だ。でも、体のトレーニングばかりじゃ頭の方がなまってしまう。頭のトレーニング

さっそくだが前回の解答に入ろう。「2-D RUBIK」はやさしかったとみえて、ほとんどの人があっていた。正解の手数と、その手順の1例を示しておく。

- NO.1 2手(2-5)  
NO.2 3手(3-5-1)  
NO.3 4手(3-5-1-5)  
NO.4 3手(6-3-5)  
NO.5 4手(1-6-3-5)

次に、「とりかえっこはずる」だが、これはむずかしすぎたようだ。46手という正解を書いてきた人はほとんどいなかった。図のようにワクに番号を付けると、正解は次のとおりになる。

図

1	2	3		
4	5	6		
7	8	9	10	11
		12	13	14
		15	16	17

1. 10→9 2. 8→10 3. 7→8  
4. 9→7 5. 12→9 6. 6→12  
7. 3→6 8. 9→3 9. 15→9  
10. 16→15 11. 10→16 12. 8→10  
13. 9→8 14. 11→9 15. 14→11  
16. 12→14 17. 6→12 18. 5→6  
19. 8→5 20. 2→8 21. 1→2

22. 7→1 23. 9→7 24. 11→9  
25. 17→11 26. 16→17 27. 10→16  
28. 13→10 29. 12→13 30. 6→12  
31. 4→6 32. 7→4 33. 9→7  
34. 10→9 35. 8→10 36. 2→8  
37. 3→2 38. 9→3 39. 15→9  
40. 12→15 41. 6→12 42. 9→6  
43. 11→9 44. 10→11 45. 8→10  
46. 9→8

ところが、この「46手」が合っていた数名の人も、こんなにむずかしいパズルが解けたにもかかわらず、2-D RUBIK パズルで間違えていて、両方とも正解の者は“ナン”。とても残念だ。今回の問題に期待しよう。

### CAT & MOUSE PUZZLE

やあ、ボクはいたずらネコの“チュー”だ。ネコなのに、チューなんておかしい名前だと思うだろう。これにはフカーイワケがあるんだ。あれはたしか1年前のちょうど今ごろだった……なんて話してるひまはないので、次に行こう。

ボクの悪友にネズミの“ニャン”がいるんだ。ネズミなのに、ニャンという名前がついたのにはフカーイワケがある。あれはたしか……やめておこう。

名前がおかしくても、やっぱりネコとネズミ。いつも追いかけてっこをくり返している。そこできょうは、きみがボクのかわりにネズミのニャンのやつをつかまえてほしいんだ。

プログラムを走らせると分かるけど、五角形の頂点と中心に計6個の門があり、線でつながれた門にだけ移動できる。先手はきみだ。行きたい門の番号を押してごらん。ネコがピッと動くから。次はニャンの番。こいつはもちろん、ネコから逃げるように動く。今度はまたきみの番だ。こうして交互にひとつずつ動いて、ニャンのやつをつかまえばいいんだ(分かてると思うけどニャンはネズミだよ)。自分の番には必ず動かなくちゃいけない。「動かないよ！」ってすわり込んだじゃだめ。

ニャンをつかまえられるようになったら、今度はなるべく早くつかまえてほしい。きみが動くとき1手、ネズミが動くとき2手、というように数えて、なるべく少ない手数でおねがいする。さあ、ガンバって！



# CAT & MOUSE PUZZLE

画面を見れば分かると思うけど、T字のくしは、片側には最高でも中、小のダンゴ1個ずつの計2個しか入れられないし、たとえば大のダンゴ1個入れれば、もうその側には別のダンゴは1個も入れられないんだ。また、T字のくしの片方だけ太くなっていて、小のダンゴは通せないようになってる。さて、操作の仕方だが、カーソル移動キーの左右のボタンを使って、どちら向きに動くのかを指定すると、動く先に自分自身が入れるスペースのあるダンゴがその方向に動く。もし、その方向に動けるダンゴがふたつある場合には"WHICH?"(どちら?)と聞いてくるので、動かしたいダンゴが向かって左右のどちらにあるかをやはりカーソ

ル移動キーで指定する。また、スペースを押せば、まん中のT字型をしたくしが180度回転する。左右3つのダンゴを入れ替えられるようになったら、今度はまたいつもの「最少手数捜し」にとりかかってほしい。まん中のT字型のくしの回転も1手と数えるから、気をつけてね。

正解者のうち抽選で5名様に、好きなMSXソフトを差し上げます。ふるってご応募ちょーだい(正解をタノムゼ!!)。宛先はもう暗記した人もいると思う。

〒102 東京都千代田区四番町2-1  
 株日本ソフトバンク 出版部  
 「Oh!HiTBit」パズル係

画面を見れば分かると思うけど、T字のくしは、片側には最高でも中、小のダンゴ1個ずつの計2個しか入れられないし、たとえば大のダンゴ1個入れれば、もうその側には別のダンゴは1個も入れられないんだ。また、T字のくしの片方だけ太くなっていて、小のダンゴは通せないようになってる。さて、操作の仕方だが、カーソル移動キーの左右のボタンを使って、どちら向きに動くのかを指定すると、動く先に自分自身が入れるスペースのあるダンゴがその方向に動く。もし、その方向に動けるダンゴがふたつある場合には"WHICH?"(どちら?)と聞いてくるので、動かしたいダンゴが向かって左右のどちらにあるかをやはりカーソ

ル移動キーで指定する。また、スペースを押せば、まん中のT字型をしたくしが180度回転する。左右3つのダンゴを入れ替えられるようになったら、今度はまたいつもの「最少手数捜し」にとりかかってほしい。まん中のT字型のくしの回転も1手と数えるから、気をつけてね。

正解者のうち抽選で5名様に、好きなMSXソフトを差し上げます。ふるってご応募ちょーだい(正解をタノムゼ!!)。

宛先はもう暗記した人もいると思う。

〒102 東京都千代田区四番町2-1

株日本ソフトバンク 出版部  
 「Oh!HiTBit」パズル係

## CAT & MOUSE PUZZLE

```

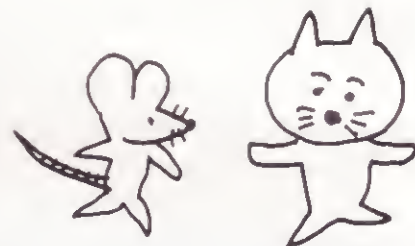
10 *****
20 ' cat & mouse
30 *****
40 DIM D(6),K(6,3),G(2)
50 OPEN"GRP:" FOR OUTPUT AS #1
60 COLOR 15,1,7
70 SCREEN 2,2
80 FOR I=1 TO 16
90 READ X1,Y1,X2,Y2
100 LINE(X1,Y1)-(X2,Y2),3
110 NEXT I
120 DATA 119,30,52,78,137,30,71,78,119,30,186,78,137,30,204,78
130 DATA 55,78,81,157,67,78,93,157,189,78,163,157,201,78,175,157
140 DATA 87,152,169,152,87,162,169,162,123,30,123,100,133,30,133,100
150 DATA 121,100,80,157,135,100,94,157,121,100,162,157,135,100,176,157
160 FOR I=1 TO 6
170 READ X,Y:CIRCLE(X,Y),20,4
180 PAINT (X,Y),4,4
190 NEXT I
200 DATA 128,100,128,30,195,78,169,157,87,157,61,78
210 DRAW"BM16,15":PRINT#1,"CAT-CH THE"
220 DRAW"BM152,15":PRINT#1,"MOUSE PUZZLE"
230 FOR I=1 TO 6
240 READ X,Y:SU$=RIGHT$(STR$(I),1)
250 PSET(X,Y),1
260 PRINT#1,SU$
270 NEXT I
280 DATA 125,124,125,1,219,75,194,154,59,154,34,75
290 FOR J=1 TO 2
300 A$=""
310 FOR I=1 TO 32
320 READ B$
330 A$=A$+CHR$(VAL("&H"+B$))
340 NEXT I
350 SPRITE$(J)=A$
360 NEXT J
370 DATA 06,0A,12,22,23,40,64,84,E4,80,E0,80,71,20,19,07,60,50,48,44,C4,02,26,21
,27,01,07,02,8E,04,98,E0
380 DATA 00,3C,42,82,87,88,90,A2,40,40,46,58,E6,1C,67,80,00,3C,42,41,E1,11,09,45
,06,02,62,1A,67,38,E6,01
390 FOR I=1 TO 16
400 READ X,Y,S
410 K(X,Y)=S
420 NEXT I
430 DATA 1,1,2,1,2,4,1,3,5,2,1,1,2,2,3,2,3,6,3,1,2,3,2,4,4,1,1,4,2,3,4,3,5,5,1,1
,5,2,4,5,3,6,6,1,2,6,2,5
440 C=1:M=3
450 H=1:GOSUB 950
460 PUT SPRITE 1,(X,Y),14,1
470 H=3:GOSUB 950
480 PUT SPRITE 2,(X,Y),14,2
490 LINE(220,179)-(250,179),14
500 GOTO 780
510 *****
520 FOR I=1 TO 6:D(I)=1:NEXT I
530 D(C)=0
    
```



```

540 IF C=1 THEN D(6)=2:D(3)=2:GOTO 600
550 IF C=2 THEN D(5)=2:D(4)=2:GOTO 600
560 IF C=3 THEN D(1)=2:D(5)=2:D(6)=2:GOTO 600
570 IF C=4 THEN D(2)=2:D(6)=2:GOTO 600
580 IF C=5 THEN D(2)=2:D(3)=2:GOTO 600
590 IF C=6 THEN D(1)=2:D(3)=2:D(4)=2:GOTO 600
600 IF M=3 OR M=6 THEN ZZ=2 ELSE ZZ=3
610 L=0
620 FOR I=1 TO ZZ
630 IF D(K(M,I))=2 THEN L=L+1:G(L)=K(M,I)
640 NEXT I
650 IF L=0 THEN 750
660 IF L=2 THEN M=G(INT(RND(1)*2)+1):GOTO 680
670 M=G(1)
680 FOR Q=1 TO 300:NEXT Q
690 PUT SPRITE 2,(0,0),4,0
700 H=M:GOSUB 950
710 PUT SPRITE 2,(X,Y),14,2
720 BEEP
730 GOSUB 1100
740 GOTO 780
750 E=1
760 IF D(K(M,E))=1 THEN M=K(M,E):GOTO 680
770 E=E+1:GOTO 760
780 '
790 A$=INKEY$:IF A$="" THEN 780
800 IF VAL(A$)>0 AND VAL(A$)<7 THEN 810 ELSE 790
810 IF C=3 OR C=6 THEN ZZ=2 ELSE ZZ=3
820 GO=0
830 FOR I=1 TO ZZ
840 IF VAL(A$)=K(C,I) THEN GO=1
850 NEXT I
860 IF GO=0 THEN 1010
870 C=VAL(A$)
880 PUT SPRITE 1,(0,0),4,0
890 H=C:GOSUB 950
900 PUT SPRITE 1,(X,Y),14,1
910 GOSUB 1100
920 BEEP
930 IF C=M THEN 1020
940 GOTO 520
950 IF H=1 THEN X=120:Y=92:RETURN
960 IF H=2 THEN X=120:Y=22:RETURN
970 IF H=3 THEN X=187:Y=70:RETURN
980 IF H=4 THEN X=161:Y=149:RETURN
990 IF H=5 THEN X=79:Y=149:RETURN
1000 IF H=6 THEN X=52:Y=70:RETURN
1010 GOTO 780
1020 PUT SPRITE 2,(0,0),4,0
1030 FOR I=1 TO 30
1040 X=INT(RND(1)*110)+40
1050 Y=INT(RND(1)*191)
1060 BEEP
1070 PSET(X,Y),1:PRINT#1,"ツカマシタ!!"
1080 NEXT I
1090 GOTO 1090
1100 LINE(250,178)-(220,170),1,BF
1110 KA=KA+1:PRINT#1,KA
1120 RETURN

```



## K U R U K U R U P U Z Z L E

```

10 '*****
20 ' FURU FURU PUZZLE
30 '*****
40 DIM C(11),T(6),F(2)
50 OPEN"GRP:" FOR OUTPUT AS #1
60 COLOR 15,4,7
70 SCREEN 2
80 LINE(0,130)-(255,191),1,BF
90 LINE(83,90)-(128,96),10,BF
100 LINE(128,91)-(173,95),10,BF
110 LINE(123,80)-(133,130),12,BF
120 DRAW"BM65,30":PRINT#1,"FURU FURU PUZZLE"
130 LINE(62,38)-(190,58),14
140 MM=2:K=184:GOSUB 1410
150 K=232:GOSUB 1460
160 K=204:GOSUB 1540
170 MM=1:K=72:GOSUB 1410
180 X=24:GOSUB 1460
190 X=52:GOSUB 1540
200 LINE(8,91)-(80,95),10,BF
210 LINE(176,91)-(148,95),10,BF
220 LINE(0,80)-(7,130),12,BF
230 LINE(244,80)-(255,130),12,BF
240 FOR I=1 TO 5
250 READ XX,R
260 CIRCLE (XX,150),R,3
270 PAINT (XX,150),3,3
280 CIRCLE (XX+125,150),R,8
290 PAINT (XX+125,150),8,8

```



```

200 NEXT I
210 DATA 4.3,2.50,4.57,3
220 FOR I=1 TO 3
230 READ XX,R
240 CIRCLE (XX,150),R,8
250 PLOT (X,150),8,8
260 CIRCLE (X+125,150),R,3
270 PLOT (X+125,150),3,3
280 NEXT I
290 DATA 73.3,80,4,90,6
300 DRAW "S4BM110,145C14R30U5F10G10U5L30U10"
310 PLOT (111,146),14,14
320 FOR I=1 TO 5 STEP 2
330 T(I)=4-(I+1)/2:T(I+1)=7+(I+1)/2
340 NEXT I
350 FOR I=1 TO 10
360 READ Z(I)
370 DATA 4,3,2,0,0,0,0,2,3,4
380 NEXT I
390 F(1)=1:F(2)=0
400 *****
410 ON STICK(0) GOTO 510,510,540,510,510,510,930,510
420 IF STRIG(0)=-1 THEN 1810
430 GOTO 510
440 ML=0:MR=0
450 FOR JJ=1 TO 6
460 IF T(JJ)<4 AND Z(T(JJ)+1)=0 THEN 600
470 IF T(JJ)=7 AND Z(8)=0 THEN 620
480 IF T(JJ)=6 AND Z(7)=0 AND Z(8)=0 THEN 620
490 GOTO 540
500 IF 5-Z(4)-Z(5)<Z(T(JJ)) THEN 640
510 ML=JJ:GOTO 540
520 IF 9-Z(1)-Z(2)-Z(10)<Z(T(JJ)) THEN 640
530 MR=JJ
540 NEXT JJ
550 IF T(ML)>2 AND Z(T(ML))=2 AND F(1)=1 THEN ML=0
560 IF ML=0 AND MR=0 THEN 510
570 IF ML=0 THEN MM=MR:GOTO 700 ELSE IF MR=0 THEN MM=ML:GOTO 700
580 GOTO 1730
590 *****
600 IF T(MM)=5 THEN 740
610 IF T(MM)=7 THEN 870
620 X=(Z(1)+Z(2)+Z(3)-Z(T(MM)))/2+8
630 ON MM GOSUB 1630,1630,1690,1690,1660,1660
640 IF Z(5)<>0 THEN 900
650 SWAP Z(T(MM)),Z(5):T(MM)=5
660 X=123-Z(5)/2+8
670 ON MM GOSUB 1410,1410,1540,1540,1460,1460
680 GOTO 860
690 X=133+(Z(6)/2)+8
700 ON MM GOSUB 1630,1630,1690,1690,1660,1660
710 X=248-(Z(8)+Z(9)+Z(10)+Z(T(MM))/2)+8
720 ON MM GOSUB 1410,1410,1540,1540,1460,1460
730 IF Z(10)=0 THEN SWAP Z(10),Z(T(MM)):T(MM)=10:GOTO 860
740 IF Z(9)=0 THEN SWAP Z(9),Z(T(MM)):T(MM)=9:GOTO 860
750 SWAP Z(8),Z(T(MM)):T(MM)=8
760 GOTO 1310
770 X=133+(Z(6)+Z(7)/2)+8
780 ON MM GOSUB 1630,1630,1690,1690,1660,1660
790 GOTO 810
800 SWAP Z(T(MM)),Z(4):T(MM)=4
810 X=123-(Z(5)+Z(4)/2)+8:GOTO 770
820 *****
830 ML=0:MR=0
840 FOR JJ=1 TO 6
850 IF T(JJ)>7 AND Z(T(JJ)-1)=0 THEN 940
860 IF T(JJ)=4 AND Z(3)=0 THEN 1010
870 IF T(JJ)=5 AND Z(4)=0 AND Z(3)=0 THEN 1010
880 GOTO 1030
890 IF 5-Z(6)-Z(7)<Z(T(JJ)) THEN 1030
900 MR=JJ:GOTO 1030
910 IF 9-Z(1)-Z(2)-Z(3)<Z(T(JJ)) THEN 1030
920 ML=JJ
930 NEXT JJ
940 IF T(MR)<>0 AND Z(T(MR))=2 AND F(2)=1 THEN MR=0
950 IF ML=0 AND MR=0 THEN 510
960 IF ML=0 THEN MM=MR:GOTO 1090 ELSE IF MR=0 THEN MM=ML:GOTO 1090
970 GOSUB 1730
980 *****
990 IF T(MM)=5 THEN 1180
1000 IF T(MM)=4 THEN 1260
1010 X=248-(Z(8)+Z(9)+Z(10)-Z(T(MM))/2)+8
1020 ON MM GOSUB 1630,1630,1690,1690,1660,1660
1030 IF Z(6)<>0 THEN 1290
1040 SWAP Z(T(MM)),Z(6):T(MM)=6
1050 X=133+Z(6)/2+8
1060 ON MM GOSUB 1410,1410,1540,1540,1460,1460
1070 GOTO 1250
1080 X=123-(Z(5)/2)+8
1090 ON MM GOSUB 1630,1630,1690,1690,1660,1660
1100 X=8+(Z(1)+Z(2)+Z(3)+Z(T(MM))/2)+8
1110 ON MM GOSUB 1410,1410,1540,1540,1460,1460
1120 IF Z(1)=0 THEN SWAP Z(1),Z(T(MM)):T(MM)=1:GOTO 1250
1130 IF Z(2)=0 THEN SWAP Z(2),Z(T(MM)):T(MM)=2:GOTO 1250
1140 SWAP Z(3),Z(T(MM)):T(MM)=3
1150 GOTO 1310

```





```

1200 X=123-(Z(5)+Z(4)/2)*8
1270 ON MM GOSUB 1030,1630,1690,1690,1660,1660
1280 GOTO 1200
1290 SWAP Z(T(MM)),Z(7):T(MM)=7
1300 X=133+(Z(6)+Z(7)/2)*8:GOTO 1160
1310 LINE(123,80)-(133,130),12,BF
1320 GOSUB 2190
1330 IF T(1)=8 AND T(3)=9 AND T(5)=10 THEN 1340 ELSE 1390
1340 IF T(2)=3 AND T(4)=2 AND T(6)=1 THEN 1350 ELSE 1390
1350 DRAW"BM95,65":PRINT#1,"MADE IT !! "
1360 DRAW"BM60,181":PRINT#1,"PLAY AGAIN ?(Y/N)"
1370 A$=INKEY$:IF A$="" THEN 1370
1380 IF A$="N" OR A$="n" THEN END ELSE IF A$="Y" OR A$="y" THEN RUN ELSE 1370
1390 GOTO 510
1400 *****
1410 IF MM=INT(MM/2)*2 THEN C=8 ELSE C=3
1420 CIRCLE(X,93),8,C
1430 PAINT (X,93),C,C
1440 LINE(X-8,91)-(X+8,95),10,BF
1450 RETURN
1460 IF MM=INT(MM/2)*2 THEN C=8 ELSE C=3
1470 CIRCLE(X,93),16,C,.4,2,8
1480 LINE(X-15,86)-(X+15,88),C
1490 PAINT (X,78),C,C
1500 CIRCLE(X,93),16,C,3,5,5,95
1510 LINE(X-15,98)-(X+15,98),C
1520 PAINT (X,99),C,C
1530 RETURN
1540 IF MM=INT(MM/2)*2 THEN C=8 ELSE C=3
1550 CIRCLE(X,93),12,C,.55,2,7
1560 LINE(X-11,88)-(X+11,88),C
1570 PAINT (X,82),C,C
1580 CIRCLE(X,93),12,C,3,7,5,8
1590 LINE(X-11,98)-(X+11,98),C
1600 PAINT (X,99),C,C
1610 RETURN
1620 *****
1630 LINE(X-8,85)-(X+8,90),4,BF
1640 LINE(X-8,103)-(X+8,96),4,BF
1650 RETURN
1660 LINE(X-16,77)-(X+16,89),4,BF
1670 LINE(X-16,109)-(X+16,97),4,BF
1680 RETURN
1690 LINE(X-12,109)-(X+12,97),4,BF
1700 LINE(X-12,77)-(X+12,89),4,BF
1710 RETURN
1720 *****
1730 DRAW"BM103,65":PRINT#1,"WHICH ?"
1740 ON STICK(0) GOTO 1740,1740,1760,1740,1740,1740,1770,1740
1750 GOTO 1740
1760 MM=MR:GOTO 1780
1770 MM=ML:GOTO 1780
1780 LINE(103,65)-(170,71),4,BF
1790 RETURN
1800 *****
1810 SWAP Z(4),Z(7):SWAP Z(5),Z(6)
1820 FOR J=1 TO 6
1830 IF T(J)=4 THEN T(J)=7:GOTO 1870
1840 IF T(J)=7 THEN T(J)=4:GOTO 1870
1850 IF T(J)=5 THEN T(J)=6:GOTO 1870
1860 IF T(J)=6 THEN T(J)=5:GOTO 1870
1870 NEXT J
1880 SWAP F(1),F(2)
1890 LINE(83,77)-(122,109),4,BF
1900 LINE(134,77)-(173,109),4,BF
1910 IF F(1)=1 THEN 1950
1920 LINE(83,91)-(128,95),10,BF
1930 LINE(128,90)-(173,96),10,BF
1940 GOTO 1970
1950 LINE(83,90)-(128,96),10,BF
1960 LINE(128,91)-(173,95),10,BF
1970 LINE(123,80)-(133,130),12,BF
1980 FOR MM=1 TO 6
1990 IF T(MM)<>5 THEN 2030
2000 X=123-Z(5)/2*8
2010 ON MM GOSUB 1410,1410,1540,1540,1460,1460
2020 GOTO 2140
2030 IF T(MM)<>4 THEN 2070
2040 X=123-(Z(5)+Z(4)/2)*8
2050 ON MM GOSUB 1410,1410,1540,1540,1460,1460
2060 GOTO 2140
2070 IF T(MM)<>6 THEN 2110
2080 X=133+Z(6)/2*8
2090 ON MM GOSUB 1410,1410,1540,1540,1460,1460
2100 GOTO 2140
2110 IF T(MM)<>7 THEN 2140
2120 X=133+(Z(6)+Z(7)/2)*8
2130 ON MM GOSUB 1410,1410,1540,1540,1460,1460
2140 NEXT MM
2150 LINE(123,80)-(133,130),12,BF
2160 GOSUB 2190
2170 GOTO 510
2180 *****
2190 LINE(150,180)-(116,171),1,BF
2200 CO=CO+1:PRINT#1,CO
2210 RETURN

```







# MSX MUSIC EDITOR II

○伴奏の種類 (1~7)

○アルペジオ部分のオクターブ (1~6)

○ベース部分のオクターブ (1~6)

を尋ねてきますから、この場合、順番に、4, 1, 5, 1を入力してみてください。すると、ほら!! 伴奏付きで聞こえてきたでしょう。

もう一度 **[F1]** を押して、こんどは4, 3, 3, 1を入力してみましょう。伴奏の感じが変わったでしょう? 伴奏の種類や、音の高さを変えて、どれが一番ぴったりするか、試してみてください。

ただし、伴奏の種類で、6か7を指定すると、なんだか変になってしまいます。この6と7は、3拍子の伴奏用なので、ワルツ・イング・マチルダには向きません (なお、コードは1小節にふたつまでです)。

## 記憶する

さて、この2小節についてはこれでOK、ということになったら、**[F2]** を押します。すると1~20の番号が表示されます。これがレコードナンバーです。現在表示されている音楽のデータを記憶しておくときに、1~20の記憶場所があると考えてください。

今、REC NO? と表示されているところで、1を入力すると、**[F1]** を押したときと同じように、オクターブや伴奏の種類を尋ねられるはず。そして音楽が流れますね。この音楽が、NO.1に記憶されるのです。

さて、楽譜を見れば分かるように、この曲では、1, 2小節と5, 6小節、そして13, 14小節がまったく同じです。けれども曲の流れを考えると、伴奏の感じだけは変えたいですね。

そこで、もう一度 **[F2]** を押します。今度はREC NO. に3を入力します。あれ? 気がつきませんでしたか? 1~20の数字が表示されたとき、1の次に\*があったでしょう。これは、さっきNO.1に記憶したので、このナンバーは記憶済みだということ

図3

キー	意味
RETURN	音符を描くときなど
<b>[b]</b>	フラットの音符を描くとき
<b>[#]</b>	シャープの "
<b>[1]</b>	♪16
<b>[2]</b>	♪16.
<b>[3]</b>	♪8
<b>[4]</b>	♪8.
<b>[5]</b>	♪4
<b>[6]</b>	♪4.
<b>[7]</b>	♪2
<b>[8]</b>	♪2.
<b>[9]</b>	♪1
<b>[DEL]</b>	音符を消す(1つだけ)
<b>[SPACE]</b>	休符の表示
<b>[F1]</b>	play
<b>[F2]</b>	record
<b>[F3]</b>	play back
<b>[F4]</b>	SAVE
<b>[F5]</b>	code input

を示しているんです。もし、もう一度NO.1を指定すると、新しいデータが記憶されて前のは消えてしまいます。

さあ、NO.に3を指定した後はさっきと同じです。でも伴奏の種類やオクターブは変えてみましょう。同じように、NO.7にも伴奏をかえて記憶させます。

## つなげて聴く

**[F3]** を押して、1 $\odot$ 1 $\odot$ 3 $\odot$ 7 $\odot$ e $\odot$ と入力してみましょう。

すると、さっき記憶させた音楽がつながって流れます。NO.1が2回、NO.3, NO.7というように。

もちろん、入力を7 $\rightarrow$ 3 $\rightarrow$ 1 $\rightarrow$ eとすれば、その順序で流れます。

## できあがり

では、図2を参考にして、他の部分も記憶させてください。

新しい主旋律を入力するときには、**[DEL]**

を押して全部の音符を消してからでもいいですし、今表示されているのを気にせずになら書き換えていってもかまいません。また、3小節目のように、ひとつの小節にコードがひとつしかない場合でも、C $\rightarrow$ Cと2つ入力してください。

全部記憶し終わったら、また **[F3]** を押して、1~8までつなげて聴いてみます。

## データをセーブする

OK? それでは、データレコーダをセットして (録音ボタンを押して)、**[F4]** を押します。すると、それまでのデータが、BASICのデータ文になって書き込まれます。

プログラムは終了しないので、このままエディットを続けることができますが、今は**[CTRL]+[STOP]**でブレイクして、セーブしたデータをロードしてみましょう。

テープを巻き戻して、  
LOAD"CAS:pldata"

とします。OKが出たら、リストを出してみましょう。5100~5020がREC NO.1のデータです。5100に主旋律、5110と5120に伴奏のデータがあります。NO.2のデータは5200~5220行です。

これに1~210行を付け加えたのが、リスト2です。

## チャンチャン♪

「ワルツィング・マチルダ」は、初めての人でも1時間ちょっとあればできあがるでしょう。1度コードなどを入力したら、伴奏の仕方を変えて、いろいろ試してみてください。

リスト3に、松田聖子 (皆さんがこれを読む頃にはもう結婚しているでしょう) の、「ガラスのりんご」を出しておきました。でも、このデータをせつせと入力するよりも、自分で作って欲しいな。ちなみに、私はこれを作るのに、伴奏の種類など迷いながら、1時間ちょっとかかりました。みなさんも、名作・大作は、取り組んでみてください。きっと、大満足。

図3

小節	REC INC.	主せ ん律	主せん律の オクターブ	伴奏の種類	アルペジオの オクターブ	ベースの オクターブ
1.2	1	①	4	1	5	1
3.4	2	②	4	1	5	1
5.6	3	①	4	1	5	1
7.8	4	③	4	3	5	1
9.10	5	④	5	2	4	1
11.12	6	⑤	5	2	4	1
13.14	7	①	4	4	5	1
15.16	8	③	4	4	5	1

☆注:

9-12小節は、実際の楽譜より1オクターブ下げた楽譜を入力し、オクターブ入力の際に1オクターブ上げること調整している(高い音までしか表示できないので)。

☆注意:  
本来の楽譜には無いところに休符が入っています。  
これは、音がつながってしまうのを避けるためです。

## リスト1

```

10 *
20 *   MSX MUSIC EDITOR Ver.2
30 *
40 *   by JUN
50 *
60 *
1000 CLEAR 5000
1010 KEY 1,CHR$(20):KEY 2,CHR$(21):KEY 3,CHR$(22):KEY 4,CHR$(23):KEY 5,CHR$(24)
1020 DIM Y(40),L$(40),D(20),F(40),R$(40),M$(40),N(40),DAT$(12,40),CD$(14),CD(12)
1030 HL$(20),HB$(20)
1035 *
1040 FOR I=1 TO 14
1050   READ CD$(I)
1060 NEXT
1070 DATA " C","Db"," D","Eb"," E"," F","Gb"," G","Ab"," A","Bb"," B","++"
1080 SCREEN 0:KEY OFF:COLOR 15,7
1090 GOSUB 1600
1100 X=1:Y=1
1110 LOCATE X*2,20-Y:AS=INPUT$(1)
1120 A=INSTR(CHR$(28)+CHR$(29)+CHR$(30)+CHR$(31)+CHR$(13)+CHR$(127)+CHR$(20)+CHR
3(21)+CHR$(22)+CHR$(23)+CHR$(24)+"b# 123456789",A$)
1130 IF A=0 THEN 1110
1140 IF A=15 THEN IF Y(X)=0 THEN 1110
1150 ON A GOSUB 1170,1190,1210,1230,1310,1250,1770,2000,2190,2390,2560,1350,1390
1160 GOTO 1110
1170 IF X<10 THEN X=X+1
1180 IF X=10 THEN X=X-1
1190 IF X=1 THEN X=X-1
1200 IF X=10 THEN X=X-1
1210 IF Y<10 THEN Y=Y+1
1220 RETURN
1230 IF Y>1 THEN Y=Y-1
1240 RETURN
1250 *
1260 XX=X*2:GOSUB 1650
1270 Y(X)=0
1280 L$(X)=" ":GOSUB 1550
1290 F(X)=0
1300 RETURN
1310 XX=X*2:GOSUB 1650
1320 LOCATE X*2,20-Y:PRINT"●":Y(X)=Y
1330 F(X)=0
1340 RETURN
1350 XX=X*2:GOSUB 1650
1360 LOCATE X*2,20-Y:PRINT"●":Y(X)=Y
1370 LOCATE X*2,19-Y:PRINT "b":F(X)=-1
1380 RETURN
1390 XX=X*2:GOSUB 1650
1400 LOCATE X*2,20-Y:PRINT"●":Y(X)=Y
1410 LOCATE X*2,19-Y:PRINT"#":F(X)=1
1420 RETURN
1430 XX=X*2:GOSUB 1650
1440 LOCATE X*2,13:PRINT"l":Y(X)=-1
1450 RETURN
1460 L$(X)="16 ":GOSUB 1550:RETURN
1470 L$(X)="16 ":GOSUB 1550:RETURN
1480 L$(X)=" 8 ":GOSUB 1550:RETURN
1490 L$(X)=" 8 ":GOSUB 1550:RETURN
1500 L$(X)=" 4 ":GOSUB 1550:RETURN
1510 L$(X)=" 4 ":GOSUB 1550:RETURN
1520 L$(X)=" 2 ":GOSUB 1550:RETURN
1530 L$(X)=" 2 ":GOSUB 1550:RETURN
1540 L$(X)=" 1 ":GOSUB 1550:RETURN
1550 *
1560 LOCATE X*2,21 :PRINT MID$(L$(X),1,1):
1570 LOCATE X*2,22 :PRINT MID$(L$(X),2,1):
1580 LOCATE X*2,23 :PRINT MID$(L$(X),3,1):
1590 RETURN
1600 *   goson
1610 FOR XX=0 TO 39
1620   GOSUB 1650
1630 NEXT
1640 RETURN
1650 *
1660 FOR YY=0 TO 20 STEP 2
1670   LOCATE XX,YY
1680   PRINT " ":
1690 NEXT
1700 LOCATE XX,19 :PRINT " ":
1710 FOR YY=9 TO 17 STEP 2
1720   LOCATE XX,YY
1730   PRINT "- ":
1740 NEXT

```



# MSX MUSIC EDITOR II

```
1750 RETURN
1760 '
1770 '~~~~~ play ~~~~
1780 '
1790 LOCATE 10,1:INPUT "123-7" 1-6:OTS$
1800 OCT=INSTR("123456",OTS$)
1810 IF OCT=0 THEN 1790
1820 RESTORE 1870
1830 FOR Y=1 TO 18
1840 READ D(Y)
1850 D(Y)=D(Y)+OCT*12
1860 NEXT
1870 DATA 0,2,4,5,7,9,11,12,14,16,17,19,21,23,24,26,28,29,31,33,35
1880 '
1890 ES=""
1900 FOR X=1 TO 18
1910 IF Y(X)>0 THEN IF MID$(L$(X),3,1)="" THEN ES=ES+"L"+MID$(L$(X),1,2)+"N"+
STR$(D(Y(X))+F(X))+", " ELSE ES=ES+"L"+L$(X)+"N"+STR$(D(Y(X))+F(X))
1920 IF Y(X)=-1 THEN ES=ES+"R"+MID$(L$(X),1,2)
1930 NEXT
1940 GOSUB 2930
1950 PLAY ES,BB$,LL$
1960 LOCATE 10,1:PRINT SPACES$(20)
1970 X=1:Y=4:LOCATE 2,18
1980 RETURN
1990 '
2000 '~~~~~ record ~~~~~
2010 '
2020 LOCATE 0,1
2030 FOR I=1 TO 20
2040 PRINT USING "##! ":I;M$(I);
2050 IF I=10 THEN PRINT
2060 NEXT
2070 LOCATE 2,4:INPUT "REC NO ? ":RC$
2080 REC=INSTR("1234567891011121314151617181920",RC$)
2090 IF REC=0 THEN 2070
2100 REC=VAL(RC$)
2110 IF REC<1 OR REC>20 THEN 2070
2120 LOCATE 0,1:PRINT SPACES$(90):PRINT SPACES$(100):
2130 GOSUB 1770
2140 M$(REC)="*"
2150 R$(REC)=ES:RL$(REC)=LL$:RB$(REC)=BB$
2160 X=1:Y=4:LOCATE 2,18
2170 RETURN
2180 '
2190 '~~~~~ play back ~~~~~
2200 '
2210 K=1
2220 LOCATE 1,0:PRINT "PLAY BACK ";
2230 LOCATE K*3 MOD 36,(K*12)+1:LINE INPUT OS$
2240 NO=INSTR("1234567891011121314151617181920e",OS$)
2250 IF NO=0 THEN 2230
2260 IF OS$="E" OR OS$="e" THEN 2310
2270 NO=VAL(OS$)
2280 IF NO<1 OR NO>20 THEN 2230
2290 N(K)=NO:K=K+1
2300 GOTO 2230
2310 '
2320 FOR KK=1 TO K-1
2330 PLAY R$(N(KK)),RL$(N(KK)),RB$(N(KK))
2340 NEXT
2350 LOCATE 0,0:PRINT SPACES$(80);
2360 LOCATE 0,1:PRINT SPACES$(80);
2370 X=1:Y=4:LOCATE 2,18
2380 RETURN
2390 '
2400 '~~~~~ save ~~~~~
2410 '
2420 OPEN "cas:pldata" FOR OUTPUT AS #10K
2430 FOR I=1 TO 20
2440 RR$=""
2450 FOR K=1 TO LEN(R$(I))
2460 AS=MID$(R$(I),K,1)
2470 IF AS<>" " THEN RR$=RR$+AS$
2480 NEXT
2490 PRINT #1,5000+I*100;" DATA ";RR$
2500 PRINT #1,5000+I*100+10;" DATA ";RL$(I)
2510 PRINT #1,5000+I*100+20;" DATA ";RB$(I)
2520 NEXT
2530 CLOSE
2540 RETURN
2550 '
2560 '~~~~~ code ~~~~~
2570 '
2580 LOCATE 3,0
2590 FOR I=1 TO 7:PRINT CD$(I);":NEXT
2600 LOCATE 3,1
2610 FOR I=8 TO 13:PRINT CD$(I);":NEXT
2620 X=1:K=1
2630 IF X<=7 THEN LOCATE X*3,0 ELSE LOCATE (X-7)*3,1
2640 AS=INPUT$(1)
2650 IF AS=CHR$(28) AND X<13 THEN X=X+1
2660 IF AS=CHR$(29) AND X>1 THEN X=X-1
2670 IF AS<>CHR$(13) THEN 2630
2680 IF X=13 THEN 2730
2690 CD(K)=X
2700 GOSUB 2790
2710 K=K+1
2720 IF K=9 THEN 2730 ELSE 2630
2730 LOCATE 0,0:PRINT SPACES$(80);
2740 LOCATE 1,5:PRINT SPACES$(40);
2750 CD(K+1)=0:CD(K+2)=0
2760 X=1:Y=4:LOCATE 2,18
2770 RETURN
```

```
2780 '
2790 LOCATE 1,7:PRINT SPACES$(60);
2800 IF CD(K)=13 THEN 2830
2810 LOCATE 1,5:INPUT "1: MAJOR 2:MINOR 3:SEVENTH ";AS$
2820 IF INSTR("123",AS$)=0 THEN 2810
2830 C3(K)=VAL(AS$)
2840 LOCATE 1,7
2850 FOR KK=1 TO K
2860 PRINT CD$(CD(KK));
2870 IF C3(KK)=1 THEN PRINT " ";
2880 IF C3(KK)=2 THEN PRINT "m ";
2890 IF C3(KK)=3 THEN PRINT "7 ";
2900 NEXT
2910 RETURN
2920 '
2930 '~~~~~ code play ~~~~~
2940 '
2950 LOCATE 8,2:PRINT "
2960 LOCATE 8,2:INPUT "a~t ";AS$
2970 AP=INSTR("1234567",AS$)
2980 IF AP=0 THEN 2960
2990 LOCATE 8,2:INPUT "7~^~t 123-7 ";AS$
3000 COCT=INSTR("123456",AS$)
3010 IF COCT=0 THEN 2990
3020 LOCATE 8,2:INPUT " ^~s 123-7 ";AS$
3030 BOCT=INSTR("123456",AS$)
3040 IF BOCT=0 THEN 3020
3050 '
3060 BB$="":LL$=""
3070 FF=1
3080 IF CD(FF)=0 THEN 3150
3090 C1=CD(FF)-1: C3=C3(FF)
3100 IF CD(FF+1)=0 THEN C2=12: C4=C3 ELSE C2=CD(FF+1)-1: C4=C3(FF+1)
3110 GOSUB 3190
3120 LL$=LL$+LOW$
3130 BB$=BB$+BAS$
3140 FF=FF+2:GOTO 3080
3150 LOCATE 8,2:PRINT SPACES$(30);
3160 RETURN
3170 '
3180 '
3190 '
3200 CC(1,1)=0:CC(2,1)=0:CC(3,1)=0
3210 CC(1,2)=4:CC(2,2)=3:CC(3,2)=4
3220 CC(1,3)=7:CC(2,3)=7:CC(3,3)=7
3230 CC(1,4)=12:CC(2,4)=12:CC(3,4)=10
3240 CC(1,5)=16:CC(2,5)=15:CC(3,5)=16
3250 '
3260 '
3270 ON AP GOSUB 3290,3530,3780,4000,4250,4370,4600
3280 RETURN
3290 '
3300 '~~~~~ 1 ~~~~~
3310 '
3320 RESTORE 3490
3330 '
3340 LOW$="L8"
3350 FOR I=1 TO 8
3360 READ JJ
3370 IF I<=4 THEN CD=C1 ELSE CD=C2
3380 IF I<=4 THEN CH=C3 ELSE CH=C4
3390 LOW$=LOW$+"N"+STR$(CC(CH,JJ)+COCT*12+CD)
3400 NEXT
3410 '
3420 BAS$="L5"
3430 FOR I=1 TO 4
3440 IF I<=2 THEN CD=C1 ELSE CD=C2
3450 IF I<=2 THEN CH=C3 ELSE CH=C4
3460 BAS$=BAS$+"N"+STR$(CC(CH,I)+BOCT*12+CD)+"R20"
3470 NEXT
3480 '
3490 DATA 1,2,3,4,5,4,3,2
3500 '
3510 RETURN
3520 '
3530 '~~~~~ 2 ~~~~~
3540 '
3550 RESTORE 3730
3560 LOW$="L8"
3570 FOR I=1 TO 8
3580 READ JJ
3590 IF I<=4 THEN CD=C1 ELSE CD=C2
3600 IF I<=4 THEN CH=C3 ELSE CH=C4
3610 LOW$=LOW$+"N"+STR$(CC(CH,JJ)+COCT*12+CD)
3620 NEXT
3630 '
3640 RESTORE 3740
3650 BAS$="L4"
3660 FOR I=1 TO 2
3670 READ JJ
3680 IF I=1 THEN CD=C1 ELSE CD=C2
3690 IF I=1 THEN CH=C3 ELSE CH=C4
3700 BAS$=BAS$+"N"+STR$(CC(CH,JJ)+BOCT*12+CD)+"R4"
3710 NEXT
3720 '
3730 DATA 1,3,2,3,1,2,3,4
3740 DATA 1,2
3750 '
3760 RETURN
3770 '
3780 '~~~~~ 3 ~~~~~
3790 '
3800 RESTORE 3960
3810 '
```





# MSX SOUND EDITOR

本田 武 一

## はじめまして

Oh! HiTBiT の読者の皆さん、こんにちは！MSX を使うようになって、まだ半年の若葉マークです。とはいっても、ボクが初めてマイコンを手にしたのはワンボードの頃、つまり5～6年前(すごいだろ！すごくない。ガックリ)。その頃は、マシン語しか使えないで10万円、BASICを走らせるにはさらに数万円、カラーなどは夢の夢でした。外国製のリングなど床の間の飾りでした。

ここ数年ほど、マイコンから遠ざかっていたのですが、友だちからソニーのHB-75を借りて、日進月歩、時進日歩、分進日歩、秒進分歩(くどいけど、本当に進歩が早い)で、おいてけぼりをくった隔世の感を受けました。

さてMSXの特徴として、SPRITE、SCREENが4種、16色カラー、そして音などがありますが、Oh! HiTBiT誌上において「GRAPHICS EDITOR」「MUSIC EDITOR」そして「SPRITE EDITOR」などは、すべて発表済みです。今回は、残る最後の城として効果音作りには欠かせないSOUND文をいかに使うか? 「SOUND EDITOR」を作ってみました。

## イントロダクション・オブ・SOUND

ここで、SOUND文について簡単に解説をしておきます。昔懐かしい「インペダ」のミサイル音や「シーソージャンプ」の風船の破裂音、また近日発売が予定されている「MSX バンゲリングベイ」のヘリコプタ音などは、すべてSOUND GENERATORによって出されます。わがMSXには、自分で好きな音を作れるようにとPROGRAMMABLE SOUND GENERATOR、略して

PSGが搭載されています。このPSGに直接命令を伝えるのがSOUND文です。

PSGには16個のレジスタがあるのですがSOUND 2, 10(2番のレジスタに10を入れる)などのようにするわけです。実際には、0～13までのレジスタの値を操作します。

## サウンドエディタの使い方

まずプログラムをRUNさせると、画面に操作盤が表示されます。使用するキーは、カーソルキー、**[F1]**、**[F5]**、およびスペースキーです。上下にカーソルキーを押すとメニューの矢印が動きますので、試しに「SOUND 2 VOLUME」に合わせてみてください。このSOUND 1, 2, 3というのは3つの合成音が出せるということです。なお、表示の関係上、命令文に使われているSOUND文のレジスタ番号とは一致していませんのでくれぐれも御注意を。

次に、左右のキーを押して緑のレバーを16に合わせてください。そして、同じ要領で「SOUND 2 TONE」を300に合わせてみます。ここで注意することは、TONE=音程は数が多い、つまり右へいくほど低くなっていきます。次に、「SOUND ON OFF」に矢印を合わせ、下の緑のカーソルを左右に動かし、出す音と出さない音を決めます。「S2」にカーソルをもっていった、スペースキーを押します。ONは赤色になっていきます。

次は、「ENVELOPE CYCLE」です。エンベロープとは、音に強弱の変化などをつけるエフェクタのようなもので、UFOのフワフワした感じなども出せます。サイクルを2500にします。最後にパターンを選びます。左側の中段にしてみましょう(このときも左右のカーソルキーで緑のわくを移動

させてください)。さて、いよいよ音出します。**[F1]**を押してください。何か音がたたくような音がしますね。「S-1 VOLUME」を16にして、サイクルを10000にすると足音のようにも聞こえます。後は自分で色々やってみてください(なお、調整ごと**[F1]**を押してください。さもないと音色はもとのままです)。

もし、自分の好みの音が見つかったら**[F5]**を押してください。SOUND文が表示されます。このままSOUND文をセーブするには、まず「NEW」をしてから、「INS」により、頭に任意の行番号を入れてリターンキーを押し、テープなら「SAVE"CAS:(ファイル名)"」、ディスクなら「SAVE"(ファイル名)"、A」とし、アスキーセーブし、MERGEによりほかのプログラムと合体させるか、または、紙にでも写して(わずか14個です)おくのもよいでしょう。それから、エンベロープが作用するのは、ボリュームを16にしたところだけです。逆にかけたくないときは15以下にしておけばよいのです。画面下のパターン表は、よく音と照らし合わせてください。左上のパターンは、エンベロープが解除された状態を表すもので、音声が続いて出るわけではありません。とにかく、なかなか思った音は作れないものなので、習うより慣れろで、数をこなしてみてください。

## 最後に

プログラムを作るにあたり、大谷氏の作られた「コンピュータ文字ルーチン」を使わせていただきました。また、私も参考にしたのですが、もっと詳しく知りたい方は、朝日新聞社刊の「MSX 早わかり事典」を御覧になることをお勧めいたします。

```

1000 *****
1010 MSX SOUND EDITOR
1020 SE=5 :M1=1 :Y1=1
1030 *****
1040 CLEAR 800:DIM C$(8),T(3),SO(13)
1050 COLOR 17,1:SCREEN 3
1060 DEF JGR=CHR$
1070 DS=0:PO=0
1080 IF 1=0:G1=RE+5 :ON
1090 G1=2:GOSUB 3380,...,3:10
1100 ***** :M1=1 :Y1=1
1110 RESTORE 3930
1120 FOR Z1=1 TO 3:
1130 READ C$(Z1)
1140 NEXT Z1
1150 ***** :M1=1 :Y1=1
1160 FOR Z1=1 TO 9
1170 H$=""
1180 IF Z1<9 THEN RESTORE 1250
1190 FOR Z2=1 TO 8
1200 READ H
1210 H$=H$+CHR$(A)
1220 NEXT Z2
1230 SP$ITE$(Z1)=H$
1240 NEXT Z1
1250 DATA 224,160,160,64,64,160,160,224
1260 DATA 0,0,8,12,14,255,0,0
1270 ***** :M1=1
1280 Y1=0:Y1=0:M1$="MSX SOUND EDITOR":C=3:GOSUB 3700
1290 Y1=10:Y1=20:M1$="SOUND 1 VOLUME":C=5:GOSUB 3700
1300 Y1=30:M1$="SOUND 1 TONE":C=4:GOSUB 3700
1310 Y1=40:M1$="SOUND 2 VOLUME":C=5:GOSUB 3700
1320 Y1=50:M1$="SOUND 2 TONE":C=4:GOSUB 3700
1330 Y1=60:M1$="SOUND 3 VOLUME":C=5:GOSUB 3700
1340 Y1=70:M1$="SOUND 3 TONE":C=4:GOSUB 3700
1350 Y1=80:M1$="NOISE TONE":C=13:GOSUB 3700
1360 Y1=90:M1$="SOUND":C=10:GOSUB 3700
1370 Y1=98:Y1=90:M1$="ON":C=6:GOSUB 3700
1380 Y1=92:Y1=90:M1$="OFF":C=15:GOSUB 3700
1390 Y1=10:Y1=100:M1$="ENVELOPE CYCLE":C=9:GOSUB 3700
1400 Y1=110:M1$="ENVELOPE PATTERN":C=11:GOSUB 3700
1410 ***** :M1=1
1420 FOR Z1=23 TO 103 STEP 10
1430 IF Z1=93 THEN 1480
1440 LINE(125,Z1)-(205,Z1),15
1450 FOR Z2=125 TO 205 STEP 5
1460 LINE(Z2,Z1-1)-(Z2,Z1+1),15
1470 NEXT Z2
1480 NEXT Z1
1490 X1=120:Y1=90:M1$="S1 S2 S3 N1 N2 N3":C=15:GOSUB 3700
1500 ***** :M1=1 :Y1=1
1510 FOR Z1=122 TO 162 STEP 20
1520 FOR Z2=12 TO 182 STEP 85
1530 LINE(Z2,Z1)-(Z2,Z1+15)
1540 LINE(Z2,Z1+15)-(Z2+65,Z1+15)
1550 NEXT Z2
1560 NEXT Z1
1570 LINE(13,137)-(17,129),7:++
1580 LINE(97,122)-(112,137),7:++0~3,9
1590 LINE(112,137),7
1600 LINE(182,137)-(197,122),7:++4~7,15
1610 LINE(197,137),7
1620 LINE(247,137),7
1630 FOR Z=12 TO 62 STEP 15:++8
1640 LINE(Z,142)-(Z+15,157),7
1650 LINE(Z+15,142),7
1660 NEXT Z
1670 LINE(77,147),7
1680 LINE(97,142)-(112,157),7:++10
1690 LINE(112,142),7
1700 LINE(142,157),7
1710 LINE(157,142),7
1720 LINE(162,147),7
1730 LINE(182,142)-(197,157),7:++11
1740 LINE(197,142),7
1750 LINE(247,142),7
1760 FOR Z=12 TO 62 STEP 15:++12
1770 LINE(Z,177)-(Z+15,162),7
1780 LINE(Z+15,177),7
1790 NEXT Z
1800 LINE(77,171),7
1810 LINE(97,177)-(112,162),7:++13
1820 LINE(112,162),7
1830 LINE(182,177)-(197,162),7:++14
1840 LINE(197,162),7
1850 LINE(227,162),7
1860 LINE(242,177),7
1870 LINE(247,171),7
1880 ***** :M1=1
1890 Y1=210:Y1=20:M1$="00":C=5:GOSUB 3700
1900 Y1=30:M1$="0000":C=4:GOSUB 3700
1910 Y1=40:M1$="00":C=5:GOSUB 3700
1920 Y1=50:M1$="0000":C=4:GOSUB 3700
1930 Y1=60:M1$="00":C=5:GOSUB 3700
1940 Y1=70:M1$="0000":C=4:GOSUB 3700
1950 Y1=80:M1$="00":C=13:GOSUB 3700
1960 Y1=100:M1$="00000":C=9:GOSUB 3700
1970 LINE(120,98)-(134,98),2
1980 LINE(10,120)-(80,139),2,8
1990 ***** :M1=1

```



```

2000 V(1)=0:V(2)=0:V(3)=0:T(1)=0:T(2)=0:T(3)=0:NT=0:S1=0:S2=0:S3=0:N1=0:N2=0:N3=
0:EC=0:EP=0:YJ=20:EU=120:DX=10:DY=120:DA$=""
2010 FOR Z=1 TO 8
2020 Y=Z*10+9
2030 IF Y=89 THEN Y=99
2040 PUT SPRITE Z,(124,Y),12
2050 NEXT Z
2060 GOTO 2110
2070 ST=STICK(0)
2080 IF ST=0 GOTO 2130
2090 IF ST=1 THEN YJ=YJ-10:IF YJ=10 THEN YJ=110
2100 IF ST=5 THEN YJ=YJ+10:IF YJ=120 THEN YJ=20
2110 PUT SPRITE 9,(0,YJ-1),9
2120 JM=(YJ-10)/10
2130 IF ST=3 OR ST=7 OR JM=8 THEN ON JM GOSUB 2160,2320,2160,2320,2160,2320,2500
,2810,2630,3100
2140 FOR Z=1 TO 100:NEXT Z
2150 GOTO 2070
2160 "+++++ オビ 1,2,3 ノ 1177 77"
2170 VB=(JM+1)/2
2180 ST=STICK(0)
2190 DMY=V(VB)
2200 IF ST=1 OR ST=5 THEN 2210 ELSE 2230
2210 IF V(VB)=16 AND EP=0 THEN EX=10:EY=120:LINE(DX,DY)-(DX+70,DY+19),1,B:LINE(E
X,EY)-(EX+70,EY+19),2,B
2220 RETURN
2230 IF ST=3 THEN V(VB)=V(VB)+1:IF V(VB)=17 THEN V(VB)=16
2240 IF ST=7 THEN V(VB)=V(VB)-1:IF V(VB)=-1 THEN V(VB)=0
2250 PUT SPRITE JM,(V(VB)*5+124,JM*10+9),12
2260 IF DMY=V(VB) THEN 2310
2270 IF V(VB)<10 THEN M1$=" 0"+RIGHT$(STR$(V(VB)),1):GOTO 2290
2280 M1$=STR$(V(VB))
2290 LINE(234,JM*10+10)-(249,JM*10+18),1,BF
2300 X1=226:Y1=JM*10+10:C=5:GOSUB 3700
2310 GOTO 2180
2320 "+++++ オビ 1,2,3 ノ 777 77"
2330 TB=JM/2
2340 ST=STICK(0)
2350 DMY=T(TB)
2360 IF ST=1 OR ST=5 THEN RETURN
2370 IF ST<>3 THEN SA=0
2380 IF ST<>7 THEN SB=0
2390 IF ST=3 THEN SA=(SA+1)*2-1:T(TB)=1(TB)+SA:IF T(TB)/4095 THEN T(TB)=4095
2400 IF ST=7 THEN SB=(SB+1)*2-1:T(TB)=T(TB)-SB:IF T(TB)<0 THEN T(TB)=0
2410 PUT SPRITE JM,(T(TB)/51.2+124,JM*10+9),12
2420 IF DMY=T(TB) THEN 2490
2430 IF T(TB)<10 THEN M1$=" 000"+RIGHT$(STR$(T(TB)),1):GOTO 2470
2440 IF T(TB)<100 THEN M1$=" 00"+RIGHT$(STR$(T(TB)),2):GOTO 2470
2450 IF T(TB)<1000 THEN M1$=" 0"+RIGHT$(STR$(T(TB)),3):GOTO 2470
2460 M1$=STR$(T(TB))
2470 LINE(218,JM*10+10)-(249,JM*10+18),1,BF
2480 X1=210:Y1=JM*10+10:C=4:GOSUB 3700
2490 GOTO 2340
2500 "+++++ ノイス ノ 777 77"
2510 ST=STICK(0)
2520 DMY=NT
2530 IF ST=1 OR ST=5 THEN RETURN
2540 IF ST=3 THEN NT=NT+1:IF NT=32 THEN NT=31
2550 IF ST=7 THEN NT=NT-1:IF NT=-1 THEN NT=0
2560 PUT SPRITE JM,(NT*2.5+124,JM*10+9),12
2570 IF DMY=NT THEN 2620
2580 IF NT<10 THEN M1$=" 0"+RIGHT$(STR$(NT),1):GOTO 2600
2590 M1$=STR$(NT)
2600 LINE(234,JM*10+10)-(249,JM*10+18),1,BF
2610 X1=226:Y1=JM*10+10:C=13:GOSUB 3700
2620 GOTO 2510
2630 "+++++ イッ"ロ-77" ノ 777 77"
2640 ST=STICK(0)
2650 DMY=EC
2660 IF ST=1 OR ST=5 THEN RETURN
2670 IF ST<>3 THEN SA=0
2680 IF ST<>7 THEN SB=0
2690 IF ST=3 THEN SA=(SA+1)*2-1:EC=EC+SA:IF EC/65535 THEN EC=65535
2700 IF ST=7 THEN SB=(SB+1)*2-1:EC=EC-SB:IF EC<0 THEN EC=0
2710 PUT SPRITE 8,(EC/819.2+124,JM*10+9),12
2720 IF DMY=EC THEN 2800
2730 IF EC<10 THEN M1$=" 0000"+RIGHT$(STR$(EC),1):GOTO 2780
2740 IF EC<100 THEN M1$=" 000"+RIGHT$(STR$(EC),2):GOTO 2780
2750 IF EC<1000 THEN M1$=" 00"+RIGHT$(STR$(EC),3):GOTO 2780
2760 IF EC<10000 THEN M1$=" 0"+RIGHT$(STR$(EC),4):GOTO 2780
2770 M1$=STR$(EC)
2780 LINE(210,JM*10+10)-(249,JM*10+18),1,BF
2790 X1=202:Y1=JM*10+10:C=9:GOSUB 3700
2800 GOTO 2640
2810 "+++++ 7777" ノ 77 77 77"
2820 ST=STICK(0)
2830 O$=INKEY$
2840 DMY=EU:UU=EU
2850 IF O$=" " THEN ON (EU-120)/24+1 GOSUB 2960,2980,3000,3020,3040,3060
2860 IF ST=0 GOTO 2820
2870 IF ST=1 OR ST=5 THEN RETURN
2880 IF ST=3 THEN EU=EU+24:IF EU=264 THEN EU=120
2890 IF ST=7 THEN EU=EU-24:IF EU=96 THEN EU=240
2900 IF DMY=EU THEN 2930
2910 LINE(UU,98)-(UU+14,98),1
2920 LINE(EU,98)-(EU+14,98),2
2930 FOR Z=1 TO 200:NEXT Z
2940 GOTO 2820
2950 "+++++ ノイス ノ 777 77"
2960 IF S1=0 THEN S1=1 ELSE S1=0

```



```

3070 M1$="S1":C=(S1+15)-S1+8:GOTO 3080
3080 IF S2=0 THEN S2=1 ELSE S2=0
3090 M1$="S2":C=(S2+15)-S2+8:GOTO 3080
3100 IF S3=0 THEN S3=1 ELSE S3=0
3110 M1$="S3":C=(S3+15)-S3+8:GOTO 3080
3120 IF N1=0 THEN N1=1 ELSE N1=0
3130 M1$="N1":C=(N1+15)-N1+8:GOTO 3080
3140 IF N2=0 THEN N2=1 ELSE N2=0
3150 M1$="N2":C=(N2+15)-N2+8:GOTO 3080
3160 IF N3=0 THEN N3=1 ELSE N3=0
3170 M1$="N3":C=(N3+15)-N3+8:GOTO 3080
3180 M1=EU:Y1=90:GOSUB 3700
3190 RETURN
3200 ***** 10000000000000000000
3210 ST=ST1(CX,0)
3220 DY=EP
3230 IF ST=0 GOTO 3110
3240 IF ST=1 OR ST=5 THEN 3250
3250 IF ST=3 THEN EP=EP+1:IF EP=9 THEN EP=0
3260 IF ST=7 THEN EP=EP-1:IF EP=-1 THEN EP=8
3270 IF DY=EP THEN 3240
3280 IF EP<3 THEN EX=EP*85+10 :EY=120:GOTO 3210
3290 IF EP>6 THEN EX=(EP-3)*85+10:EY=140:GOTO 3210
3300 EX=(EP-3)*85+10:EY=160
3310 LINE(CX,DY)-(CX+70,DY+19),1,B
3320 LINE(EX,EY)-(EX+70,EY+19),2,B
3330 DX=EX:DY=EY
3340 GOTO 3110
3350 ***** 10000000000000000000
3360 LINE(EX,EY)-(EX+70,EY+19),9,B
3370 IF EP=0 AND V(1)=16 THEN N=1:GOSUB3310
3380 IF EP=0 AND V(2)=16 THEN N=3:GOSUB3310
3390 IF EP=0 AND V(3)=16 THEN N=5:GOSUB3310
3400 RETURN
3410 ***** 10000000000000000000
3420 PUT SPRITE N,(199,N*10+9),12
3430 LINE(234,N*10+10)-(249,N*10+18),1,BF
3440 X1=220:Y1=N*10+10:M1$=" 15":C=5:GOSUB 3700
3450 M1(N+1)/2=15
3460 GOSUB 3380
3470 RETURN
3480 ***** カウント ノ カフ
3490 SO(0)=VAL("&H"+RIGHT$(STRING$(4-LEN(HEX$(V(1))),"0")+HEX$(V(1)),2))
3500 SO(1)=VAL("&H"+LEFT$(STRING$(4-LEN(HEX$(V(1))),"0")+HEX$(V(1)),2))
3510 SO(2)=VAL("&H"+RIGHT$(STRING$(4-LEN(HEX$(V(2))),"0")+HEX$(V(2)),2))
3520 SO(3)=VAL("&H"+LEFT$(STRING$(4-LEN(HEX$(V(2))),"0")+HEX$(V(2)),2))
3530 SO(4)=VAL("&H"+RIGHT$(STRING$(4-LEN(HEX$(V(3))),"0")+HEX$(V(3)),2))
3540 SO(5)=VAL("&H"+LEFT$(STRING$(4-LEN(HEX$(V(3))),"0")+HEX$(V(3)),2))
3550 SO(11)=VAL("&H"+RIGHT$(STRING$(4-LEN(HEX$(V(3))),"0")+HEX$(V(3)),2))
3560 SO(12)=VAL("&H"+LEFT$(STRING$(4-LEN(HEX$(EC)), "0")+HEX$(EC),2))
3570 SO(7)=S3-(S1+1)-(S2+2)-(S3+4)-(N1+8)-(N2+16)-(N3+32)
3580 IF EP=1 THEN SO(13)=0
3590 IF EP=2 THEN SO(13)=4
3600 IF EP=3 THEN SO(13)=8
3610 IF EP=8 THEN SO(13)=EP-8
3620 SO(6)=INT(SO(8)/V(1)):SO(9)=V(2):SO(10)=V(3)
3630 FOR Z=0 TO 13
3640 SOUND Z,SO(Z)
3650 NEXT Z
3660 IF V(1)<16 AND V(2)<16 AND V(3)<16 THEN PLAY"V8" ELSE 3600
3670 FOR Z=0 TO 5
3680 SOUND Z,SO(Z)
3690 NEXT Z
3700 RETURN
3710 ***** テキスト ラ タス
3720 COLOR 15,4,7:SCREEN 0
3730 FOR Z=0 TO 13
3740 DA$=DA$+" SOUND"+STR$(Z)+", "+STR$(SO(Z))
3750 IF Z<13 THEN DA$=DA$+";"
3760 NEXT Z
3770 PRINT DA$
3780 END
3790 GOTO 3690
3800 /----- テキスト ラ フル-テン
3810 / H.0920's PROGRAM
3820 / M1$ X1 Y1 C
3830 MK=LEN(M1$)
3840 FOR X=X1 TO X1+(MK-1)*8 STEP 8
3850 X2=(X-X1)/8+1
3860 M2$=MID$(M1$,X2,1)
3870 IF ASC(M2$)=32 THEN 3910
3880 IF ASC(M2$)<58 THEN CC=ASC(M2$)-47:GOTO 3800
3890 CC=ASC(M2$)-54
3900 DRAW "BM"+STR$(X)+", "+STR$(Y1)+"+C"+STR$(C)+CR$(CC)
3910 NEXT X
3920 RETURN
3930 DATA R5D6L3U3L1D3L1U5,BR3D6L1U2,D2U2R5D4U3L1D3L4D2R5,D1U1R5D3L2R2D3L5U1,BR3
3940 D2U1E4D1G1P1D4E1L5
3950 DATA R4L4D3R5D3U2L1D2L4U1,R5D1BL5D5R5U3L4D2U3,R5D6L1U2,R4D2F1D3L5U2F1U4L1D1
3960 R3,R5D6U3L5U2
3970 DATA BR2R2F1D5U3L4D3L1U2E1,R4D2F1D3L5U5R1D4U3R2,R5D1BL5D5R1U2R1D2R3U1,R4F1D
3980 46L4E1U1G1U4,R5L4D3R3L4U2D5R5
3990 DATA R5L4D3R3L4U2D5,R5D1BL5D5R1U2D2R4U3L2,D6F1U6D6R4U3D6,BR2D6R1U2,R5G1D5L1
4000 U2D2L2U1
4010 DATA D6U3R3U2E1G1D4E1D2R1,D6E1U1D2R4,D6R1U6F1D1R3U2D6,D6U6F1D1R1D2R1D1F1U6R
4020 1D6,R5D1L2D1R2D4L5U5
4030 DATA R5D3L3H1D4L1U5,R5G1D1E1D5H3F3L5U5,R5D3L1U2D2L2F3U1G1BL4U3R1L1U2,R5D1BL
4040 5D2R5D3L1U2D2L4U1,R5L3D6R1U2
4050 DATA D6R1U6D6R4U6,D5R1U5D5F1R1E2U4,D5F1E1U2R1D2F1E1U5,R1G1R1D1F1G2D1R1U1BE2
4060 E2U1B04BL1F1D1,D2F1R3E1U2B6D3D3
4070 DATA R5D1G2L1G2R5D1L5

```



# (METAMORPHOSIS)



図 1



前号に引き続き、オランダの版画家エッシャー(M. C. Escher)の世界にCGでチャレンジします。テーマは、ひとつの形が他の形へと変化していく様子を表すメタモルフォーシス(Metamorphosis)。このメタモルフォーシスを手軽に楽しんでもらうよう、エディタのプログラムを掲載したので、ぜひ皆さんもエッシャーの世界、メタモルフォーシスの世界をご自分のCGで味わってみてください。

## フロイト・エッシャー・CG

vol. 5で、シュルレアリスム(超現実派)の芸術家たちに大きな影響を与えた人物としてフロイトの名前を挙げました。このフロイトに再度登場してもらい、もう少し解説を加えて、メタモルフォーシスばかりでなく、「絵画と心」といった広い視野から話していきます。

20世紀初頭、オーストリアのウィーンで活動をしたジークムント・フロイト(Sig-

mund Freud 1856~1939)は、その生涯のほとんどを精神的な瑕疵(きず)に基づく病気の原因究明と治療法の確立に捧げ、精神分析の大綱を初めて世に示しました。フロイト以前にも神経病・神経痛など神経の病気の肉体的原因の研究(脳出血、脳膜炎、神経炎、神経衰弱、ヒステリー)は進められていましたが、(当時)原因不明の精神病については、病気のふりとか、想像病として片付けられている状態でした。フロイトは精神的な病気の症例を集め、患者の心の奥深くに閉じ込められた原因を睡眠状態、あるいはカウンセリングの中から引き出しました。そして、過去の体験や事件が人間の精神形成に重大な影響を与えていることを明らかにし、患者の精神の発展過程における精神的・肉体的な瑕疵と現在の病状を結びつけ、過去の精神的な<傷>をいやす(無意識に持っている抑圧を解除する)ことにより病気を治したり、和らげたりする

治療法を作り出したのです。さらに彼は、夢や言い間違い、物忘れといった現象を追求し、普通見過ごされがちな日常的な精神の営みの中でも個人の欲求や願望の重みが生じていることを解き明かしました。この夢とか無意識の行為を多数集めて書かれたのが「夢判断」(1900年)です。フロイトの考え方は、長い間学会や世間から認められず、その後発表された「日常生活の精神病理」「性理論に関する3つの論文」では、精神発達のプロセスと性的な欲求との関連にこだわり過ぎているという批判を受けました。第1次大戦を境にして、各方面から人間や社会のエゴイズムを考え直す機運が高まり、フロイトが始めた精神分析も見直されるようになりました。その後、フロイトの弟子たちの活躍で、個人だけでなく、政治・経済といった分野でも、精神分析的な考え方が取り入れられていきます。フロイト(や弟子たち)が示した精神分析が絵画、文学、音楽などに与えた影響はは



図2

や、知れないものがあります。  
私たちがCGを楽しんだり、絵を描いた  
りする心の奥をのぞいて見ると、単に描こ  
うとする対象の形や、形から受けるイメー  
ジを写しとりたいたか、新しいCG パター  
ンを見つけたい、といった欲求だけでなく、  
そのようにして得られた画面、モノを誰か  
に見て欲しい、持っていて欲しいという  
欲求も含まれているようです。その誰かと  
いうのは、母親であったり、友だちであつた  
り、恋人であったりするわけです。この描  
くことと、見せることとを分けて考えられ  
るようになります。描く対象は変化しても見  
せたい相手は余り変わらないのに気が付き  
ます。となると、描く対象というものは、  
何を意味は持たず何でもよいということ  
になります。たとえば、花は美しいもの  
であり、その美しいものを絵に描いて相手  
に見せたいとき、また、相手をモデルにして  
絵を描くとき、見せたい相手と画面を結び  
つけるものは何でしょうか。単純な見方を

すると、花は美しい。その美しいものを美  
しいと感じる私は美しい。花は美しい、美  
しい姿のまま画面の中に定着させたい……  
そんな私は美しい。君は美しい、その美し  
さを感じず私の感情は美しい。つまり、相  
手に見せたいのは花とかモデルになった相  
手とかの対象の形ではなく、その対象を通  
じて見た、自らの感情や思考なのです。  
CGでアニメのキャラクタを上手に描きた  
いと思うとき、アニメのキャラクタそのも  
のを描きたいと思っているのか、上手に描  
ける自分を見せたいのかは説明しなくても  
分かるでしょう。また、アニメのキャラク  
タを描く対象に選ぶのは、与えられている  
形をただ画面に置き換えるだけですから大  
変安易です。また、描き上った画面は誰で  
も知っている画面ですから、先ほどの花と  
は違って、描く本人の個性は求められませ  
ん。間違わずに写しとれば上手、途中でや  
めればそれまでですから、上手、下手はあ

りません。したがって、そこに描かれた画  
面は、絵画の世界のCGではなく、ジグソ  
ーパズルのような、ゲームに近いCGであ  
るといえるでしょう。

フロイトの考えた分析的な見方で絵画・C  
Gの描くという行為を見つめると、それは、  
自分の心を表現する、または無意識の内に  
表現してしまっている、ということに気が  
付きます。他人の絵や画面を見るときは、  
画面に表現された作者の心と対話を交して  
いるということにもなります。チョット難  
しくなりましたネ。表現に対する内的、精  
神的な意味付けは、フロイトの精神分析に  
よるとして、前回同様、美術史の中からアン  
テンボルトの画面のような、ある姿から  
他の姿への置き換えを示す絵を追ってみま  
す。

図1、2はルネサンスがヨーロッパ全域に広  
まっていたころ、スペインで独自の画面を  
創作していたヒエロニムス・ボッスの作品  
です。教会の祭壇画として描かれていると





います(1495年頃)。本当はこの絵でなく、動物の顔が次第に人の顔に変化していくデッサンを見ていただきたかったのですが、手に入りませんでしたので、似ているということでこれを選びました。動物の顔が人間の顔に変化するデッサンでは、ダ・ビンチが人間の獣性、エゴイズムを表すのに動物の形を用いました。と同時に、人間の“種の起源”を追求しようとする科学者の姿勢が強うかがえます。いずれにせよ、図3に表された人物たちの骨格には人間離れをしたところがあります。

図4は説明の必要はないですかね、12世紀平安時代に鳥羽僧正の描いた〈鳥獣戯画〉です。これは平安時代に多く描かれた絵巻もかわらず、その画面は普通の天国とか地獄の様子とは異なります。

図3は、ルネサンス発祥地イタリアの、というより、ルネサンスの運動を支えた芸術家の1人であるレオナルド・ダ・ビンチのデッサンです。この図の題名は「奇怪な顔・

物の一部ですから、この部分は見たことがなくても他の部分を教科書などできくと見ていると思います。また、このウサギやカエルが何を表すものであるかとか、物語の内容は知っていますね。

図5は、フロイトの研究を進めていた時期と余り変わらない19世紀末に描かれたクノッフのものです。図6は、やはり同時期に描かれたルドンのものです。この頃は、後期印象派の画家たちが大きな力をもっていました。同時進行の形でこの図5、6のような絵が描れていたというのは面白いものです。クノッフたち(ラッサンフォッス、モンタルド、デルヴィル、クリムト)を世紀末派と呼び、アールヌーボーの典型となる装飾性を大切にしました。この装飾性には、日本の障壁画の影響が強く見受けられます。同時期の印象派では、障壁画ではなく浮世絵版画だったのを思い出すと一層興味深いものです。

図7、8は、今回のテーマというか課題となるエッシャーのものです。図9に示すイラストは「不思議の国のアリス」挿画です。この図9は、絵画の世界の例というよりも、物語そのものが夢の世界でしたネ。

風刺画、戯画、幻想絵画といった系譜に見受けられる、この変身、変貌はなにを意味するのでしょうか。人は他人や社会に向かって何かいいたい、叫びたいといった願望を抱いていてもなかなか言葉に出せないことが多いようです。そして、もし自分がもっと強かったら、美しかったら、権力があつたらと考えたり、反対に自分はそのままにしておいて、相手をか弱いもの、醜いもの、悪いものに置き換え、自分が優位に立てたらと考えます。この想像の世界での変身願望が参考図版の中に表現されています。この変身願望の裏側にあるのが、コンプレックスであり、コンプレックスと同列にあるのが、本当の自分はこんなはずではないのだというナルシシズムです。そしてこの



図4

変身動物は、自分は努力をしなくてはならないという自意識（自己規制）とは正反対の「楽観（楽観）」逃避であることが多いようです。このような精神の状態は前回のアナモルフォーシスの描く、歪められたものの歪しさを察する、といった二律背反が現れているようです。今回のテーマ、メタモルフォーシスは、変身、変態、または、ここでもう一つ、その過程でもあります。歪められたものの歪しさを察する、といった二律背反が現れているようです。今回のテーマ、メタモルフォーシスは、変身、変態、または、ここでもう一つ、その過程でもあります。

変身動物は、自分は努力をしなくてはならないという自意識（自己規制）とは正反対の「楽観（楽観）」逃避であることが多いようです。このような精神の状態は前回のアナモルフォーシスの描く、歪められたものの歪しさを察する、といった二律背反が現れているようです。今回のテーマ、メタモルフォーシスは、変身、変態、または、ここでもう一つ、その過程でもあります。

変身動物は、自分は努力をしなくてはならないという自意識（自己規制）とは正反対の「楽観（楽観）」逃避であることが多いようです。このような精神の状態は前回のアナモルフォーシスの描く、歪められたものの歪しさを察する、といった二律背反が現れているようです。今回のテーマ、メタモルフォーシスは、変身、変態、または、ここでもう一つ、その過程でもあります。

見て安らぎを覚えるとき、自分の中にある潜在的な欲求不満に気が付きます。また、意識しないまま自身の内に様々な制約や抑圧を引き込んでいく心の歪みに気が付きます。逆に、彼らの中に、自分の理想とする姿や力を見出すこともあります。精神形成が未熟な段階では欲望や願望は生のまま、むき出しのままの姿を現し、その欲望や願望は野生の動物の欲望に近い姿をもっています。自分の理想と考えるものの姿は精神の発達と共に変化をするわけです。ということは、理想のものの変化と同じように自身も変化・変身をしていることになります。M.C. エッシャーの世界はその人や精神、欲望、夢といったものの変化する様を何段階かのステップを与えて描きました。魚から鳥へ、影から実体へ。しかし、エッシャーは、自分ではけっしてこれらの世界にのめり込むことなく、冷静に、あくまで外側に身を置いて眺めています。描かれたものも

けっして外側に出て来ようとしません。このエッシャーの外部対象を冷やかに見つめる視線の内に込められた諧謔は、ひとつひとつの生命は滅んでも人間の欲望、エゴイズム（利己的な理想・満足）は滅びない。または、夢は目が覚めれば消えてしまうということなのでしょう。

といったことをふまえた上で今回の実験を紹介しましょう。先に書いた、夢の世界→想像の世界→物語（アニメ）を実現させてみましょう。これは単に M.C. エッシャーの世界の模倣というのではなく、エッシャーが生きていたら、きっとエッシャー自身が手がけたであろうと思われる実験です。一平面上でのメタモルフォーシス（視覚的な目移りの時空間）ではなく、実際の時間を持った空間（アニメ）でのメタモルフォーシスです。つまり、ひとつの自我の形態が時と共に変化して行く様子が得られるのです。

いつものように、この辺で大谷氏にバトン





タッチをしてプログラムの説明をしていた  
だきましよう。

## METAMORPHOSIS EDITOR

メタモルフォーシス・エディタといっても、  
宮入氏やボクが皆さんの目の前で変身して  
カエルになったり、鳥になったりするわけ  
ではありません。ひと昔前までは、主に純  
粋芸術の対象であったメタモルフォーシス  
が、最近では実用の具として供されている  
ことの説明としてMICRO IN BETWEEN  
SYSTEMを作ってみたのです。

IN BETWEENとは、すなわち「中間的な  
もの」という意味で、転じてアニメーショ  
ンにおける「中割り」の生成の意味に用い  
られています。つまり、動きというものを、  
その動きを特徴づけるいくつかの場面（キ  
ーフレーム）とそれらを結びつける中間的  
な場面とに分けて考え、前者をアニメター  
が決定し、その情報をもとに後者をコン  
ピュータが決定するものです。従来手作業

に頼っていたアニメーションの省力化の中  
心となるのが、IN BETWEEN SYSTEM  
というわけです。

さてそこで、今回は既にでき上がっている  
CGを使ってその情報を取り込み、加工し  
ようとするプログラムも計算時間も膨大  
なものになってしまうので、簡易グラフィ  
ックスエディタ（線画のみ）をつけて、ふ  
たつのキーフレームの間を中割りするよう  
なプログラムを考えてみました。

基本的には、キーフレームとなる線画の情  
報（端点の座標）を配列に取り込んでおき、  
指定された中割りの数で比例計算を行い、  
中間点の出力を行っています。

では、MSXのプログラムからどうぞ。

© MSXETAMORPHOSIS (エムエスエッ  
クセタモルフォーシス)

RUNするとデータの有無を聞いてきます。  
初回では当然データはないので、Nを入力  
してください（実際には“Y”と“y”しか見

ていませんので、単にRETURNキーのみ  
でもOKです）。

スクリーンが変わり、“FIRST FRAME”  
の表示が出ると、画面中央で点滅している  
点を使ってデータの入力（といっても線画  
を描くだけで、データは自動的に取り込ま  
れます）が可能になります。

描画を開始したい地点までカーソルキーを  
使って点を移動させ、スペースバーを押す  
と始点が設定されて、以後スペースバーを  
押すたびに連続して線が引かれます。訂正  
が必要な場合は[BS]キーを押すごとに1  
ステップずつの削除も可能です。

入力データ数は画面下に表示されますが、  
このプログラムでは最大データ数を100個  
としていますので、100個入力した時点で  
“END FRAME”の入力に移ります。ただ  
し、100個以下のデータで終わることも多い  
でしょうから、その際には[1]か[0]を押  
してください。[0]の場合は最後に打った  
点を最終データとしますが、[1]ではもう



図5



図6

一度画面に戻り、それを最終データとします。同じ図形を描くときは [1] で終了すると便利です。さて、続く“END FRAME”と同様に入力しますが、データのカウンタでは異なり、“START FRAME”で入力されたデータ数からの減算で行われます。これは、中割りを作るときに“START FRAME”のデータと“END FRAME”のデータが1対1に対応していることが必要なので、結果が0になると両者に同数のデータが入れられたとみなして、次のメニューに移ります。

“END FRAME”にも同じ図形が必要となるときは、カウンタが1になった時点で [1] を押してください。また、途中で [1] や [0] を押して入力を終了することもできますが、“END FRAME”でのデータ数に従って中割りを決めますので、“START FRAME”後のデータの1画が使われないことになるためお気を付けください。

データが1画分になると、表示およびデー

タセーブのメニューが出て来るので任意のメニューを選んでください。

1と2を選ぶと表示のため中割りの数を聞いてきます。これは“START FRAME”と“END FRAME”の間隔を決めるもので、実際に描かれる図形の数、両端のFRAMEを含めて、この数+1になります。たとえば、間隔として1を入力した場合は、“START FRAME”と“END FRAME”の図形しか描かれません。

メニューの1と2との違いはスクリーンモードにあり、1ではSCREEN 2に単色(白)で、2ではSCREEN 3に透明色と背景色を除いた色を使って描かれます。

MSXではパレットを使えないので、リアルタイムでアニメーションというわけにはいきませんが、このSCREEN 3で描いたものなど、写真に撮って額におさめればちょっとしたインテリアにもなるでしょう(できれば、アナログRGB出力のものがよ





図 7

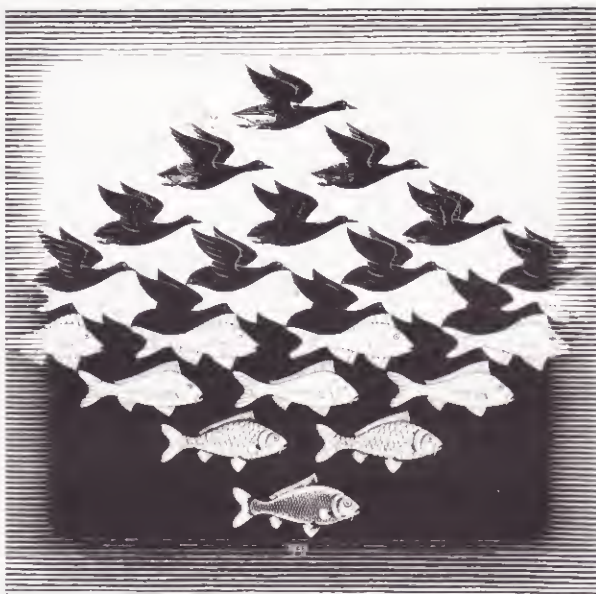


図 8

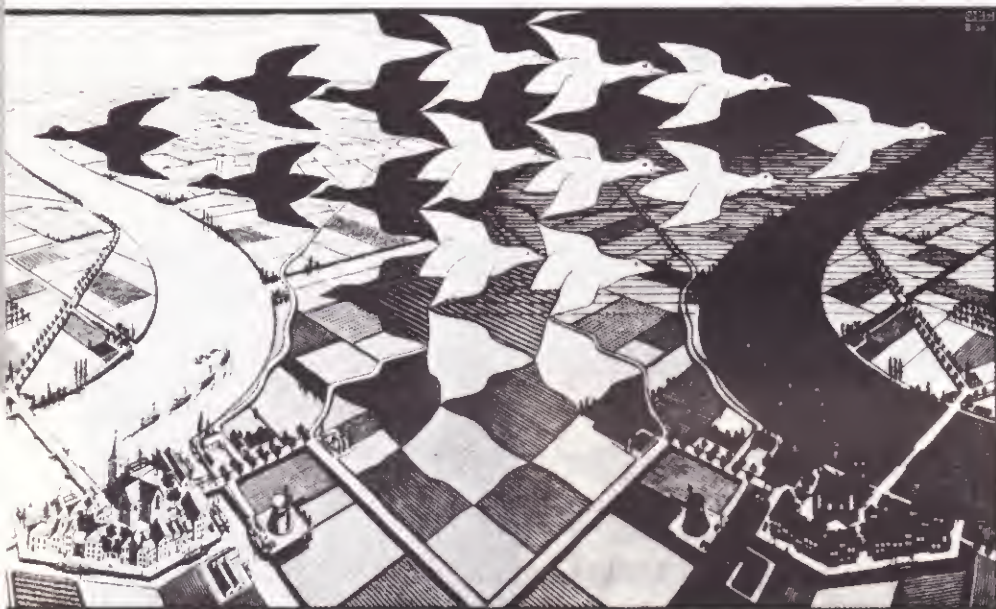


図 9



いと思いますが…。

気に入ったパターンのデータはテープやディスクにセーブしておけますから、1や2で図柄の確認をしたら、何かキーを押してメニューに戻り、3を選びます。ファイル名（ディスクならそのまま、テープの場合は頭にCAS:をつけてください）を入力した後で、装置の確認（カセットは録音状態になっていますか？など）をして“Y”を入力します。なお、ここでセーブしたデータは、プログラムの始めの“LOAD DATA, IF ANY”のところ“Y”を入力することにより、同じ手順で呼び出せます（ただし、テープを使うときは再生状態にします）。この場合にはエディタは起動せず直接表示のメニューになりますから、指示に従って

ください。

4のNEWは新しいデータを作るとき5のENDは文字どおり終了するときに使ってください。

© SMCETAMORPHOSIS (エスエムセタモルフォーシス)

使用法はMSX版とほぼ同じですから、相違点のみを説明しておきます。

まずSMC版のエディタでは、ラバーバンドカーソルが使えるので、点の位置決めが容易になっています。伸び縮みする直線で、水平・垂直の具合を見てください。

そして、これはカラーパレットボードを使っている777ユーザーと、777Cのユーザーに限られますが、表示用メニューの2がパ

レットアニメーションになっているので、形が変化していく様子をリアルタイムで確認できます。“START FRAME”と“END FRAME”の図形の位置が近いと中割りの線が重なって下になった線が消えてしまうので、気を付けてください。また、たとえば“START FRAME”を上から描き始めたとしたら、“END FRAME”は下から描き始めてみたりして、変形にヒネリを加えた面白いものを作ってみてください。

```

1000 '////////////////////////
1010 '/// METAMORPHOSIS 7"12"36 ///
1020 '/// MSX 7"07"36 ///
1030 '/// Oh!HitBit No.6 List 1 ///
1040 '/// K.Miyairi & K.Otani ///
1050 '/// 1985 Studio B.C. ///
1060 '////////////////////////
1070 '
1080 GOSUB 2000' (INITIALIZE)
1090 '
1100 GOSUB 3000' (EDIT)
1110 '
1120 GOTO 4000' (DISPLAY)
1130 '
2000 '-----
2010 'INITIALIZE
2020 '-----
2030 SCREEN 0,,0
2040 DEFINT A-Z:GOSUB 2500
2050 DIM SX(99),SY(99),EX(99),EY(99),BX(99),BY(99)
2060 X1=128:Y1=96:COUNT=1
2070 X2=128:Y2=96:CNT=0:FLG=1
2080 DIM DX(8),DY(8)
2090 DX(0)= 0:DY(0)= 0
2100 DX(1)= 0:DY(1)=-1
2110 DX(2)= 1:DY(2)=-1
2120 DX(3)= 1:DY(3)= 0
2130 DX(4)= 1:DY(4)= 1
2140 DX(5)= 0:DY(5)= 1
2150 DX(6)=-1:DY(6)= 1
2160 DX(7)=-1:DY(7)= 0
2170 DX(8)=-1:DY(8)=-1
2180 '
2190 INPUT "LOAD DATA,IF ANY (Y/N) ";ANS$
2200 IF NOT (ANS$="Y" OR ANS$="y") THEN RETURN
2210 INPUT "LOAD FILE NAME (USE CAS: FOR TAPE FILE)";F$
2230 INPUT "READY (Y/N) ";ANS$
2240 IF NOT (ANS$="Y" OR ANS$="y") THEN GOTO 2230
2250 COUNT=0
2260 OPEN F$ FOR INPUT AS #1
2270 INPUT #1,SX(COUNT),SY(COUNT),EX(COUNT),EY(COUNT)
2280 IF NOT EOF(1) THEN COUNT=COUNT+1:GOTO 2270
2290 CLOSE:CLS
2300 DFLG=1:RETURN
2500 '
2510 DIM C(13)
2520 C( 0)= 1:C( 1)=14
2530 C( 2)=15:C( 3)= 7
2540 C( 4)= 5:C( 5)=13
2550 C( 6)= 6:C( 7)= 8
2560 C( 8)= 9:C( 9)=11
2570 C(10)=10:C(11)= 3
2580 C(12)= 2:C(13)=12
2590 RETURN
3000 '-----
3010 'EDITOR
3020 '-----
3030 IF DFLG=1 THEN RETURN ELSE SCREEN 2:OPEN "GRP:" AS #1
3040 DRAW"BM16,0":PRINT #1,"START FRAME":
DRAW"BM 16,8":PRINT #1,"1)CONNECT & END 0)END or BS"
3050 GOSUB 3190
3060 FOR I=0 TO COUNT
3070 SX(I)=BX(I)
3080 SY(I)=BY(I)
3090 NEXT
3100 X1=128:Y1=96:CNT=COUNT+1
3110 X2=128:Y2=96:COUNT=1:FLG=2
3120 LINE(0,0)-(255,7),4,BF:DRAW"BM16,0":PRINT #1,"END FRAME"
3130 GOSUB 3190
3140 FOR I=0 TO COUNT
3150 EX(I)=BX(I)
3160 EY(I)=BY(I)
3170 NEXT
3180 RETURN
3190 '
3200 LINE(255,191)-(16,183),4,BF:PRINT #1,"COUNT= ";CNT
3210 X1=X1+DX(STICK(0))
3220 Y1=Y1+DY(STICK(0))
3230 PC=POINT(X1,Y1):PSET(X1,Y1),15
3240 PSET(X1,Y1),PC
3250 ST=STRIG(0):ON ST+2 GOTO 3260,3210
3260 '
3270 PSET(X1,Y1),15:BX(0)=X1:BY(0)=Y1
3280 X2=X1:Y2=Y1
3290 LINE(96,191)-(72,183),4,BF:IF FLG=1 THEN PRINT #1,COUNT
3300 IF FLG=2 THEN PRINT #1,CNT-COUNT:IF CNT-COUNT=0 THEN GOTO 3540
3310 IF STRIG(0)<>0 THEN GOTO 3310
3320 '
3330 X1=X1+DX(STICK(0))
3340 Y1=Y1+DY(STICK(0))
3350 PC=POINT(X1,Y1):PSET(X1,Y1),15
3360 PSET(X1,Y1),PC
3370 ST=STRIG(0):ON ST+2 GOTO 3390,3380
3380 K$=INKEY$:ON 1-(K$=CHR$(&H31))-2*(K$=CHR$(&H30))-3*(K$=CHR$(&H8))
GOTO 3330,3490,3540,3570
3390 '
3400 BX(COUNT)=X1
3410 BY(COUNT)=Y1
3420 LINE(X1,Y1)-(X2,Y2),15
3430 X2=X1:Y2=Y1
3440 COUNT=COUNT+1
3450 LINE(96,191)-(72,183),4,BF:IF FLG=1 THEN PRINT #1,COUNT:
IF COUNT=100 THEN GOTO 3540
3460 IF FLG=2 THEN PRINT #1,CNT-COUNT:IF CNT-COUNT=0 THEN GOTO 3540
3470 IF STRIG(0)<>0 THEN GOTO 3470
3480 GOTO 3330
3490 '
3500 BX(COUNT)=BX(0)

```



```

3510 BY(COUNT)=BY(0)
3520 LINE(BX(0),BY(0))-(X2,Y2),15
3530 RETURN
3540 /
3550 COUNT=COUNT-1
3560 RETURN
3570 /
3580 IF COUNT=1 THEN PSET(X2,Y2),4:GOTO 3190
3590 COUNT=COUNT-1
3600 X1=BX(COUNT-1)
3610 Y1=BY(COUNT-1)
3620 LINE(X1,Y1)-(X2,Y2),4:PSET(X1,Y1),15
3630 X2=X1:Y2=Y1
3640 LINE(96,191)-(72,183),4,BF:IF FLG=1 THEN PRINT #1,COUNT
3650 IF FLG=2 THEN PRINT #1,CNT-COUNT
3660 IF STRIG(0)<>0 THEN GOTO 3470
3670 GOTO 3930
4000 /-----
4010 'DISPLAY
4020 /-----
4030 SCREEN 0:CLOSE
4040 PRINT "1)DISPLAY 2)DISPLAY(MODE 3)"
4050 INPUT "3)SAVE DATA 4)NEW 5)END ":DM
4060 IF DM<1 OR DM>5 THEN CLS:GOTO 4040
4070 ON DM GOTO 4080,4080,4230,4320,4340
4080 INPUT "DIVISION NO(1<=NO<=13) ":N
4090 IF N<1 OR N>14 THEN CLS:GOTO 4080
4100 ON DM GOTO 4110,4170
4110 SCREEN 2
4120 FOR I=0 TO N
4130 PC=15:GOSUB 4360
4140 NEXT
4150 IF INKEY$="" THEN GOTO 4150
4160 GOTO 4000
4170 SCREEN 3
4180 FOR I=0 TO N
4190 PC=C(I):GOSUB 4360
4200 NEXT
4210 IF INKEY$="" THEN GOTO 4150
4220 GOTO 4000
4230 /
4240 INPUT "SAVE FILE NAME (USE CAS: FOR TAPE FILE)":F$
4250 INPUT "READY (Y/N) ":ANS$
4260 IF NOT (ANS$="Y" OR ANS$="y") THEN GOTO 4250
4270 OPEN F$ FOR OUTPUT AS #1
4280 FOR I=0 TO COUNT
4290 PRINT #1,SX(I);", ";SY(I);", ";EX(I);", ";EY(I)
4300 NEXT
4310 CLOSE:CLS:GOTO 4000
4320 /
4330 RUN
4340 /
4350 SCREEN 1,,1:END
4360 /
4370 FOR J=0 TO COUNT-1
4380 LINE(SX(J)+(EX(J)-SX(J))/N*I,SY(J)+(EY(J)-SY(J))/N*I)-(SX(J+1)+(EX(J+1)-SX(J+1))/N*I,SY(J+1)+(EY(J+1)-SY(J+1))/N*I),PC
4390 NEXT
4400 RETURN

```

## リスト2

```

1000 /-----
1010 '/// METAMORPHOSIS 7*07-56 ///
1020 '/// SMC 7*07-56 ///
1030 '/// Oh!HitBit No.6 List 2 ///
1040 '/// K.Miyairi & K.Otani ///
1050 '/// 1985 Studio B.C. ///
1060 /-----
1070 /
1080 GOSUB *INIT: (INITIALIZE)
1090 /
1100 GOSUB *EDT: (EDITOR)
1110 /
1120 GOTO *DISP: (DISPLAY)
1130 /
2000 *INIT'-----
2010 'INITIALIZE
2020 /-----
2030 WIPE:CURSOR OFF
2040 DEF INT A-Z
2050 DIM SX(99),SY(99),EX(99),EY(99),BUFEX(99),BUFY(99)
2060 X1=160:Y1=100:COUNT=1
2070 X2=160:Y2=100:CNT=0:FLG=1
2080 DIM DX(8),DY(8)
2090 DX(0)= 0:DY(0)= 0
2100 DX(1)= 0:DY(1)=-1
2110 DX(2)= 1:DY(2)=-1
2120 DX(3)= 1:DY(3)= 0
2130 DX(4)= 1:DY(4)= 1
2140 DX(5)= 0:DY(5)= 1
2150 DX(6)=-1:DY(6)= 1
2160 DX(7)=-1:DY(7)= 0
2170 DX(8)=-1:DY(8)=-1
2180 /
2190 INPUT "LOAD DATA,IF ANY (Y/N) ":ANS$
2200 IF NOT (ANS$="Y" OR ANS$="y") THEN CCLEAR:RETURN
2210 DIR:PRINT:INPUT "LOAD FILE NAME ":ANS$
2219 COUNT=0
2220 OPEN /I #1,ANS$
2230 INPUT #1,SX(COUNT),SY(COUNT),EX(COUNT),EY(COUNT)
2240 IF NOT EOF(1) THEN COUNT=COUNT+1:GOTO 2230
2250 CLOSE:CCLEAR
2260 DFLG=1:RETURN
3000 *EDT'-----
3010 'EDITOR
3020 /-----
3030 IF DFLG=1 THEN RETURN

```

```

3040 PRINT (0,0);"START FRAME 1)CONNECT & END 0)END or BS"
3050 GOSUB 3190
3060 FOR I=0 TO COUNT
3070 SX(I)=BUFY(I)
3080 SY(I)=BUFY(I)
3090 NEXT
3100 X1=160:Y1=100:CNF=COUNT+1
3110 X2=160:Y2=100:COUNT=1:FLG=2
3120 PRINT (0,0);"END FRAME 1)CONNECT & END 0)END or BS"
3130 GOSUB 3190
3140 FOR I=0 TO COUNT
3150 EX(I)=BUFY(I)
3160 EY(I)=BUFY(I)
3170 NEXT
3180 RETURN
3190 /
3200 PRINT (0,24) USING "COUNT=####",CNT;
3210 X1=X1+DX(PADDLE(0))
3220 Y1=Y1+DY(PADDLE(0))
3230 GPLOT(X1,Y1),7,4
3240 GPLOT(X1,Y1),7,4
3250 ON SHOT(0)+2 GOTO 3260,3210
3260 /
3270 BUFY(0)=X1:BUFY(0)=Y1
3280 X2=X1:Y2=Y1
3290 IF FLG=1 THEN PRINT (6,24) USING "####",COUNT;
3300 IF FLG=2 THEN PRINT (6,24) USING "####",CNT-COUNT;
3310 IF SHOT(0) THEN GOTO 3310
3320 /
3330 X1=X1+DX(PADDLE(0))
3340 Y1=Y1+DY(PADDLE(0))
3350 LINE(X1,Y1)-(X2,Y2),7,4
3360 LINE(X1,Y1)-(X2,Y2),7,4
3370 ON SHOT(0)+2 GOTO 3390,3380
3380 K$=INKEY$(2):ON 1-(K$=CHR$(&H31))-2*(K$=CHR$(&H30))-3*(K$=CHR$(&H8))
GOTO 3330,3490,3540,3570
3390 /
3400 BUFY(COUNT)=X1
3410 BUFY(COUNT)=Y1
3420 LINE(X1,Y1)-(X2,Y2),7
3430 X2=X1:Y2=Y1
3440 COUNT=COUNT+1
3450 IF FLG=1 THEN PRINT (6,24) USING "####",COUNT;:IF COUNT=100 THEN GOTO 3540
3460 IF FLG=2 THEN PRINT (6,24) USING "####",CNT-COUNT;:
IF CNT-COUNT=0 THEN GOTO 3540
3470 IF SHOT(0) THEN GOTO 3470
3480 GOTO 3330
3490 /
3500 BUFY(COUNT)=BUFY(0)
3510 BUFY(COUNT)=BUFY(0)
3520 LINE(BUFY(0),BUFY(0))-(X2,Y2),7
3530 RETURN
3540 /
3550 COUNT=COUNT-1
3560 RETURN
3570 /
3580 IF COUNT=1 THEN GOTO 3190
3590 COUNT=COUNT-1
3600 X1=BUFY(COUNT-1)
3610 Y1=BUFY(COUNT-1)
3620 LINE(X1,Y1)-(X2,Y2),0
3630 X2=X1:Y2=Y1
3640 IF FLG=1 THEN PRINT (6,24) USING "####",COUNT;
3650 IF FLG=2 THEN PRINT (6,24) USING "####",CNT-COUNT;
3660 IF SHOT(0) THEN GOTO 3650
3670 GOTO 3330
4000 *DISP'-----
4010 'DISPLAY
4020 '-----
4030 WIPE:DPLG=0
4040 PRINT "1)DISPLAY 2)ANIMATION(Need PALETTE) "
4050 INPUT "3)SAVE DATA 4)NEW 5)END ";DM
4060 IF DM<1 OR DM>5 THEN CCLEAR:GOTO 4040
4070 ON DM GOTO 4090,4080,4290,4370,4390
4080 IF DPLG=1 GOTO 4180
4090 INPUT "DIVISION NO(1<=NO<=14)=";N
4100 IF N<1 OR N>14 THEN CCLEAR:GOTO 4080
4110 GCLEAR
4120 FOR I=0 TO N
4130 FOR J=0 TO COUNT-1
4140 LINE(SX(J)+(EX(J)-SX(J))/N*I,SY(J)+(EY(J)-SY(J))/N*I)-(SX(J+1)+(EX(J+1)-
SX(J+1))/N*I,SY(J+1)+(EY(J+1)-SY(J+1))/N*I),I+1
4150 NEXT
4160 NEXT
4170 DPLG=1
4180 IF DM=1 THEN CCLEAR:LOCATE (0,0):GOTO 4040
4190 CPALETTE OFF
4200 FOR I=1 TO N+1:DEF PALETTE I,0,0,0:NEXT
4210 FOR I=0 TO N
4220 DEF PALETTE I,0,0,0
4230 DEF PALETTE I+1,15,15,15
4240 PAUSE 2
4250 NEXT
4260 DEF PALETTE N+1,0,0,0
4270 ON (INKEY$(0)=CHR$(&H20))+2 GOTO 4280,4200
4280 PALETTE RESET:CCLEAR:LOCATE (0,0):GOTO 4040
4290 /
4300 CCLEAR
4310 DIR:PRINT:INPUT "SAVE FILE NAME ":ANS$
4320 OPEN /O #1,ANS$
4330 FOR I=0 TO COUNT
4340 PRINT #1,SX(I);",";SY(I);",";EX(I);",";EY(I)
4350 NEXT
4360 CLOSE:CCLEAR:LOCATE (0,0):GOTO 4040
4370 /
4380 RUN
4390 /
4400 WIPE:END

```

# (METAMORPHOSIS)





# スーダラ栄直のMSX 音楽ソフト案内

栄 直

MSXは、もともと技術的コンピュータマニア以外を対象として作られ、いわゆるホームユースとして売られている。このホームユース、素人フレンドリという色彩からMSXには数多くの周辺機器がつながる。キーボードとディスプレイでは眼鏡をかけ青白い顔で汚れた白衣の中に脂ぎった肥満した肉体を隠しサンダルをつっかけているという、従来のコンピュータマニアが喜ぶだけなのである。陽に焼けた精悍な体軀の上にスポーツマンらしい朗らかな笑みを浮べた真黒の小さな顔を乗せコンピュータに向うMSXユーザは、無理にこまごまとしたくだらない知識を詰め込むことなしにMSXを使おうとする。だからこそ入力にはキーボード以外の何かを、出力にはディスプレイ以外の何かを、コンピュータには計算以外の何かを、要求し、その結果周辺装置が必要となるのである。こうしたユーザがMSXを使うのは実用的にかあるいはまったく遊びでか、ふたつにひとつである。が無論、後者の用途にはコンピュータ自体を遊ぶといったものは含まれない。さて、遊びでコンピュータを使うそのひとつに、電子楽器に应用するというのがある。ちと昔のことだが、コンピュータミュージックといわれる音楽

の流行ったことがあった。別に、音楽がコンピュータの形をしているわけでも、中にコンピュータがはいっているわけでも、コンピュータが原料であったわけでもなく、だからほとんどコンピュータとは無関係であったのだが、こんな風な呼び方があったのである。つまりは電子っぽいことはすべてコンピュータがやってると思い込んだ人がシンセを使った音楽をこう名付けてしまったんだよー、えーいややこしい、である。まあ、シーケンサがマイコンを搭載してたしゃねえか、などと強弁されれば、そういえなくもないかもしれないと納得してしまう。が、この類の人が存在するから、民生用デュービーが魔法ビンなんてことが起きてしまっ、やっぱりものごとには筋をとおさない。そんなこんなの話があつて、パソコンとシンセは仲がいいのである。なんとなれば、ゲームマシンには電子的な効果音がツキモノであるとか、MIDIバスなどパソコンと音楽には強いつながりがあるんだよ、などの話はできるものの、ここでは直接関係ないので割愛させていただく。それでは本稿に直接関係ある話は何かと問えば、子曰く、MSXに接続可能な音楽用周辺装置とソフトウェアの紹介であると。



## FM サウンドシンセサイザユニット

MSX-01 ヤマハ ¥19,800

ユニットコネクタ UCN-01 ヤマハ ¥7,800

FMという技術に制約するのは、音を出す部  
 である。シンセサイザユニット。MSXに  
 は「PSG (Programmable Sound Generator)」  
 なるLSIが内蔵されており、ここから音を  
 出すようになっていた。が、このLSI 聞いて  
 らればあきらかなように実に貧弱な音を  
 出せてくれる。すなわち、この石ではコン  
 コールできるパラメータが非常に少ない  
 こと。三和音までしか出すことができない  
 など。シンセサイザとはいえない音しか出  
 せない。のである。ところが、このユニット  
 では、いや、これこそシンセサイザの  
 本質。このユニットが出す音というのは  
 本当の「シンセサイザの音」、いわゆる「コ  
 ンピュータサウンド」ということになる。  
 まに、シンセサイザなる楽器はこのユニ  
 ットに鍵盤をつけたもの、と考えても間違  
 ではない。そして、このユニット、とて  
 るべき。MSXのROMパック2つぶん  
 程度の大きさしかない。昔のシンセなんざ  
 あてつけえもんだった。など懐古の情に浸  
 っているあなた、科学の進歩は軽薄短小、  
 進歩なのである。しかし、ここまで小  
 型になったのにはわけがある。そのうち最  
 ら大きな貢献をしているのがFM方式で  
 ある。これによってLSIから多  
 様な音を出すことが可能となった。この  
 FM方式で音を作っているのがFM音源  
 である。さてFM方式とはどんな方式なの  
 か。ともかくいうのは非常に難しい。か  
 ついてそれをくぐく説明するのもここで  
 はできない。ざっくりとだけいってしまえば、

ま、たとえば0Vから1Vまで入力直線  
 的に変化すると正弦波が1周期分出てくる  
 ような、そんな回路を組み合わせる音を出  
 すのだ、ということになる。この回路に正  
 弦波を加えるとどんな音がでるのか、これ  
 をさらにこの回路に入れるとどうなるのか、  
 私には分からない。MSX ユーザであって  
 心身ともに健全な皆様には、私がいわん  
 としていることすら分かっていただけないだ  
 ろうと思う。FM音源はイロイロな音を出  
 すことができるが、望んでいる音を自由に  
 得るなんてことがほとんど不可能である。  
 というのがいいかかったのだ。昔のVCA  
 がどうの、VCFだのVCOだのLFOだの  
 といっていた時代とは違うのである。当時  
 のシンセではどこをどういじればどんな音  
 が出てくるのか予想できたが、その代わり  
 音の種類が少なかった。FM音源は後者を  
 大幅に伸ばした代わりに前者を放棄したの  
 だ。だからプリセットと称していろいろな  
 音があらかじめシンセにセットしてある。  
 といったところで、このユニットに話を戻  
 そう。このユニットにも48種類の音、6種  
 類のリズムがあらかじめセットしてある。  
 これらはごくスタンダードなもので、楽器  
 メーカーが作ったものらしくなっている。  
 音、リズムなどこのユニットに内蔵されて  
 いるソフトウェアで操作でき、

### CALL MUSIC

で起動する。

このソフトウェアでは音の細かな調整はで  
 きないもののいくつかの機能が提供される。  
 たとえばポリフォニックとモノフォニック  
 部分を分けキーボードに割り付けることが  
 できる。ポリフォニックはいくつもの音程  
 の音を一辺に出せ、モノフォニックでは同  
 時にひとつの音しか出せない。これらの部  
 分はキーボードを2分割して自由に割り合

てられ、すなわちすべてをポリフォニック  
 にすることもその逆もできるのである。ポ  
 リとモノでは音の種類を変えることもでき  
 る。大事なことをいい忘れていた。このソ  
 フトは別売のキーボードと組み合わせなけ  
 ればまったく意味がない。使えない。また、  
 これだけではBASICから操作することも  
 できない。その代わり、といっはなんだ  
 が価格は安い。たった2万円である。今ど  
 き2万円なんて道に落ちていたって拾わな  
 い。見付ければ絶対拾うんだが。ま、MSX  
 をもっていれば買わない手はないだろう。  
 と思った直後、ふと思い出したことがある。  
 紹介する人間がそんなにいい加減な態度では  
 困るなんていわれても、はあ面目ない。こ  
 のFMシンセユニットはヤマハのMSX  
 なら直結できるが、他社のものには無理な  
 のである。「カートリッジコネクタ UCN-  
 01」が必要となるのだ。なんだなんだ、こ  
 れでは互換性がないじゃないか、同じ  
 MSXではないか、などと思ったらそれは  
 あなたが何か勘違いをしているのである。  
 MSXはいろんな機能をもち込んで進化して  
 いるのである。そんなわけで合わせてた  
 った27,600円。ま、こんな額、今どき落ち  
 ていたって拾わないからってこたあないか  
 しらん。このユニットは、MIDIインタフ  
 ェースも内蔵している。ってことはつまり、  
 他の電子楽器と同時に演奏したりとか、他  
 のコントローラで使うとか、そんなことも  
 できるわけだ。

## 2. ミュージックキーボード

YK-10 ヤマハ ¥29,800

YK-01 ヤマハ ¥17,800

上のユニットは単体では意味がない。だか





らこのキーボードである。これで48種類の音と6種のリズムとそれらの変化が可能なシンセシステムができあがる。これを付けてしまえばほんとにシンセである。シンセユニットにはキーボード用のコネクタがあるから、そこへキーボードの線をガシャッ、ほらもうシンセ、である。YK-10は49鍵で大きく、いわゆるフツウのキーボードサイズである。YK-01は44鍵で小さい。ポータサウンドなんかと同じ大きさである。ここはひとつ、1万ちょっとの差に迷うことなくYK-10を買って欲しい。01のは一は何となくオモチャのピアノみたいな感じがするし安っぽい。その点10は場所をとるものの、部屋に飾ってサマになる。いかにも楽器という感じはするし、すこぶるキー感もよろしい。01の上に赤い塗料がこぼれていると「あー、ここの子供がハンバーガーでも食べながらキーボードいじって、ケチャップが何かをたらしたんだな。きたねえな」と思わざるを得ないが10だと「鍵盤の上に血が。すっ、すぐ金田一君を呼んでこい。えっ、死体が見つかった？ 矢追町5丁目？」ってんでいっきに赤川ミステリである。01の上にトランプが一枚落ちていれば「まあた、ちらかしたままだ。きつく叱ってやらにやいかんな」となるし、10なら「キャッ、キャッツアイの予告状だ」とな

る。01だと「ぼくにも弾けた」、10なら「音に深みがない。なぜだ」である。キーボードの上手下手に関わらず、とりあえず、YK-10はもっていてほしい。

### 3. プレイカードセット

ZPA-01 ヤマハ ¥12,800

つまりはキーボードを弾くことができず、とはいっても演奏を楽しみたい、そんなこというんだったらLPでもCDでもテープでも聞きやいいじゃねーか、などというのは短絡的で、知的好奇心のかたまりであるところのMSXユーザは、コンピュータを使って音楽を聞きたいのである。ではどうすればよいか。陽焼けして引き締った顔から健康そうに白い歯を光らせて君はいうのである「なーんだ、MSXに演奏させればいいじゃないか」かくて必要となるのが、MSXに楽譜を入力するツール、それを演奏するプログラムである。

まさにそのために作られたのがこれである。この「プレイカードセット」はデータカードを読み取る装置とそれを使ってデータを読み演奏するプログラムとがセットになっている。まずスロットにセットする部分が

あり、ここに読み取り装置のインタフェースとプログラムROMが入っている。これにはミニプラグが付いていてそこに読み取り装置からのコネクタを差し込むのである。さて、この装置で読むのはA4ぐらいの紙の下端に磁気テープの貼ってあるカードでこの磁気テープの部分にデータが記録されているものだ。ポータサウンドにはこのリードが内蔵されていて、だからいろいろな曲のデータが売られている。

このプレイカードセットはFMシンセサイザユニットがなくとも使うことができる。PSGで演奏することができるからだ。とはいうものの、やっぱりFMシンセユニットで聞くほうがよろしい。だいたい、これにFMシンセユニットとキーボードつけりや大笑いである。曲に合わせて重ね弾きとかメロディーに合わせて伴奏が付くとか、いろいろできるわけだ。

データカードの楽譜を読んで演奏するだけというのでは、一応、CPUなる頭脳を積んでいるMSXがかわいそうである。だから、プログラムにはいろいろな機能が付いてて、テンポ、キー、パート別の音量、楽器なんぞ適当に設定できる。繰り返し演奏とか画面に鍵盤を出して表示させるとかなんとかかんとか、詳しくは買って調べていただきたい。ただ、価格が、ちとお高いような気



ミュージックキーボードYK-10



がや、演奏できミセーはそろえたのに、  
 音源は無いから、とかなんとかの  
 不満は多いだろう。

## FMシンセボックス

YRM-11 ヤマハ ¥16,800

最近最も絶大な製品である。HB-75にはカ  
 ートリッジスロットが1個しかなく、だか  
 らFMシンセユニットとプレイカードセットを  
 別に買うことができなくなってしまう。  
 そこでFMシンセボックスの登場だ。これでカー  
 トリッジスロットがふたつになった。こり  
 んで、音楽の周辺機器ってわけじゃ  
 ない、ただこれが無いと使えないから紹介  
 したい。別にFMの肩をもっているわけじ  
 ゃない、でもこれにも無理が感じられる。

## FM音色プログラム

YRM-12 ヤマハ ¥7,800

音色プログラムというのも激しい方だ  
 ころにはFM Voicing Programとなっ  
 て、より一歩FM音源で自由な音作  
 りをするためのプログラムである。音作り  
 といっても作曲ではなくて、それにはそれ  
 ぞれの音がある、音自体を作ることである。  
 これを始めて始めてFMシンセサイザとしての  
 シンセユニットとなる。画面を見ながらいろいろ  
 パラメータを変えていき、好きな音を得  
 ることができ、鍵盤で音を試すこともでき  
 るというのだ。このプログラムはROM  
 カートリッジで供給されるから、すぐ使  
 える。また豊富なコマンドをも

っていて、作ったデータをテープにセーブ  
 するとかあるいはロードする、音色データ  
 をプリンタに打ち出すなど、かなり実用的  
 にできている、といえるのかどうか。作っ  
 た音色をイチイチテープから読み出すのは  
 はっきりいってだるいし、いろいろな音を  
 出せるように作られていながら考えて変更  
 するのはむずかしいし。どこをどう変え  
 らよいかよく分かんないのである。馴れ  
 れば使えるのかもしれないが、今の私には  
 とてもむずかしい。とはいえFMシンセユ  
 ニットに始まり鍵盤、音色プログラムとあ  
 ればこりやあ完全なシンセサイザとなるか  
 ら、¥7,800という価格とも相まって、買  
 って損は無いな、と思わせる一品である。

## FMミュージックコンポーザ

YRM-15 ヤマハ ¥7,800

こいつアマサに必携、五つ星、赤丸30ラン  
 クアップ、肥満に月見草オイル、サイゴン  
 解放10年、これはROMカートリッジに入  
 ったプログラムである。コンポーザである  
 からして、データを自分で打ち込んで聞い  
 てみたい人、作曲をしている人、夜一度は  
 小便に起きてしまうあなたに最適。データ  
 を画面上の五線譜に並べていけるとい  
 うすぐれもの。入力したデータは当然のことな  
 がらテープにセーブ、ロードすることがで  
 き、また、実に細かい記述がこんなものま  
 で入れられるのかとびっくりするほどで、  
 入力できる。  
 またこのプログラム用のデータも売られて  
 いる。これはカセットで供給されているの  
 だが、読み込んだデータをいじくって遊ぶ  
 こともできるし、ただ聞くだけでも「おっ

おっ」つー感じでとてもよろしい。試し  
 にBilly Joelの「Just the Way You Are」  
 を聞いたが、こりやほんとのシンセだと  
 感じられる。ま、この辺は、さすがヤマハ  
 さん、よいしょってなところである。価格  
 も非常に手頃だ。こいつあ、遊べるぜなの  
 である。

## FMミュージックマクロ

YRM-11 ヤマハ ¥7,800

プログラマは、おうそコンピュータの周  
 辺装置と呼ばれるものすべてプログラマブ  
 ルでなくてはならぬ、と考える。この「プ  
 ログラマブル」というのは無論彼らの使っ  
 ている言語から見てということである。す  
 なわち、シンセユニットを鍵盤から使うと  
 か、既成のソフトウェアで使うとか、そん  
 なところに甘んじることは断じてできない  
 のが彼らなのである。彼らは成長すると一  
 般のユーザから尊敬と驚異と謙悪を込め  
 「真の男達」と呼ばれるようになる。真の  
 男達はハードウェアさえええられれば他に  
 何の情報もなくても、彼らの日常会話であ  
 る機械語でサクサクいじりまわすのである。  
 「真の男達」志願者の日常会話はBASIC  
 だ。だから彼ら若いプログラマはBASIC  
 で周辺装置をいじり、少しでも真の男達に  
 近づこうとする。このとき必要になるのが  
 このFMミュージックマクロというソフト  
 ウェアである。これをROMカートリッ  
 ジにセットすればBASICが拡張され、FM  
 シンセユニットをいじる命令（正確には機  
 械語サブルーチンだ、とプログラマはい  
 うだろう）が付加される。が、自分でプロ  
 グラムを組んでFMシンセユニットを使





うと思う人はコンピュータ好きの、いわゆるマニアにしかないであろうと思う。だが、しかし、このソフトが提供する大きな機能のひとつに「しゃべる」ことがある。FM シンセユニットから音声を出せるのである。これははっきりいっておもしろい。実際に「しゃべら」せてみたが、聞き取れるのだ。音声の品質はそれほどとはいえないが悪くもない。遊んでみるにはちょうどよい。と聞けばマニア以外の人の食指も動くのではないか。声が出たからどうだててんてい、という冷静な人はしょうがない。たかが遊びなんだからくだらなくてと一ゼンである。

さてこの FM ミュージックマクロで拡張された BASIC では、同時に 4 つの楽器が使える 8 種和音、リズム、音声合成、MIDI などなどいろんな操作が可能である。他にもユニークな命令、たとえばイベント割込などが用意しており、プログラマにはこれもののこれらである。

## 8. エクスパンションメモリーカートリッジ HBM-64 ソニー ¥19,800

16K しかメモリをもっていない MSX、これは実にもの悲しいものである。暗くて悲惨なものである。いや、これではいけない僕だってメジャーになってナウイ服でもきてカッコイイ靴でもはいて、海岸で肌を焼いて、かわいい彼女のひとりでも欲しい、と思うのは別に恥かしいことでもなんでもなくむしろ自然なことである。そーだ、明るく楽しく生きるためにはメモリを増やせばいいんだ、そーだそーだ、16K なんかじゃ陽の当たる世界なんかでにできないやいやい

というわけだ。そこでお勧めするのがこのメモリーカートリッジ。この中には 64 KB もの RAM が押し込まれ、これをカートリッジに差せば、あっというまに RAM 64KB である。音楽関係のソフトには RAM 32 KB 以上ないと動作しないものがあるけどこれがあれば安心。ソフトが動作しない、どうせ僕の MSX はぼろいよ、どうせ貧乏だよ、なんて世をすねて、そんな時にやさしい言葉をかけてくれた頬に傷あるお兄さんについていき、鉄砲弾として一生を終わる、なんてこともなくなる。明るい一家の必需品、人生の表街道を歩くために 1 台いかがであろう。

## 9. スーパースンセ ビクター音楽産業 ¥5,800

「真の男達」は FM シンセユニットもつけない MSX からいろいろな音を出し、それに誇りをもち楽しんでいるかもしれない。「真の男達」は優秀なハードウェアを欲しがらない。ハードウェア自体の性能に少しでも可能性がみえると、それをぎりぎりいっぱいまで、たったひとつのかたまりから何百フィートものパスタを打ってしまう料理人のようなものだ、機能拡張してしまう。ところがどっこい、MSX が手元にあるきみはそんな能力、そしてそんなことにさく時間をもっていない。だいたい真の男達は能力を拡張すること自体を愛しているのである。きみはその機能が欲しいのであるからその手段などどうでもいい。だから音で遊びたいと思えば FM シンセユニットをつけるのだし、あるいはシンセサイザをリンドラムをエミュレータを買うのだし、

オーケストラそのものを買ってしまう。とはいっても手もちのキャッシュが少ない第 2 土曜日に突然音で遊びたいと思いつき、店にかけこんだものの、そこではダイナーもアメックスもヴィザも使えず、なあんとときはどうすればいいの。などと困り果てているおぼっちゃまに、この「スーパースンセ」を勧めたい。これなら手持ちの 16 KB 以上の RAM をもつ MSX になる、カートリッジポンドで音作りができるようになるこのプログラムは、

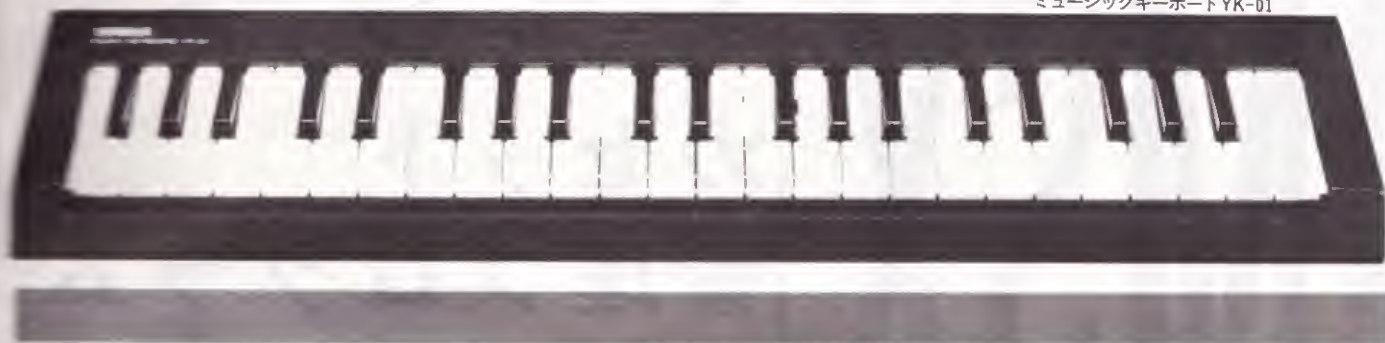
① 3 つの音を 1 組として使える  
これが短所なのか長所なのかというのは、気のもちよう次第である。まあ、人生なんてそんなもののわけで、雰囲気はダレているときにカラ元気を出していればまわりの人間にもセーキ(生気だよ)がよみ返り、なんとなく全体が活気づいて自分も元気になる、など、ちょっとたとえが長かったようである。3 つの音を 1 組として扱うと、1 音がとても厚みのある音となる。キーボードがいじれない人でも指 1 本で操作できる、ひとつの音に關していじれるパラメータが多くなっていじっていてもおもしろいということがある。

### ② 操作がカンタン

このプログラムは使い勝手がいい。画面を見ながら、スペースキーと矢じるしキーを操作するだけでたいがいのことができる。キーボードの操作がへたで、だから自分はマイコンに向かないなどという真実に気がついてしまった人も、氷水芋吉も、肥満気味の人にも、使うことができる。元首相にも勧めたい。リハビリに最適だからだ。このプログラムはトラの門病院に納入されたという実績もあるほどだ。またヒマをもて余す方にもお使いいただけるということから、そういった方の社会参加に積極的に役



ミュージックキーボードYK-01



でものもいいだろう。

メニューを用いてメニューを構成するといふのは理解しやすく使いやすくてほんとに素晴らしい。

④再生ができる

キーボードで演奏したデータが内部に貯えられる。そのデータをもとに再生することもできる。といったデータを音符にしてみることもできる。すなわち採譜ができるのである。いや、これは実際にやってみると愉快な発見も多岐、おもしろい。おもしろいおもしろいなど、やりながら心もはずみ身も動く。いやなことなどみな忘れ、しばしば笑って、などと喜ぶ自分の姿、たかがこれほど小さなことに、つついはしゃぐ自分が、思わずかし悲し情なし、人にも親にも見せられぬ。

⑤プリンタ・カセットをサポート

このプログラムはプリンタとかカセットを駆使するように作ってある。プリンタに楽譜を出力することができる。テープに音のデータや楽譜のデータをセーブロードできる。これによって作った貴重な情報を失わなくて済む。これからの情報化社会では情報が失われる構造が定着する。そんなきたるべき時代に備えた、いわば未来対応の機能とかならう。少なくともキャプテン対応のプログラムよりは将来性があるといえよう。

⑥音をこんなに変えることができる

このプログラムでは一音三重和音となっている。このために変更できるパラメータが

1. VOLUME
2. PITCH
3. NOISE
4. OCTAVE
5. VIBRATE SPEED
6. VIBRATE DEPTH
7. ECHO SPEED
8. ECHO DEPTH
9. KEY TRANSPOSE
10. SCALE
11. V3 PITCH BIAS
12. V2 PITCH BIAS
13. RELEASE TIME
14. ATTACK TIME
15. LFO SPEED
16. LEO WAVE

この16のパラメータを変更できるのである。奇数のものと偶数のものひとつずつがベアとなり奇数が横軸、偶数が縦軸の2次元グラフをいじってパラメータを変える。このグラフが8つできるわけで、絵が何となくカッコよい。機能的にどうか、などというのはどーでもいい、見た目がいいから好きだ。このうち、珍しいパラメータにスケールというやつがある。これも役に立つか

どうかは別として、とてもいい機能だ。

このプログラムでは音の人力をキーボードでやる（うーん、周辺機器の付加がないのだからトーゼンだね）のだが、このキーボードの音階が、たとえば日本陰旋法だとか全音階だとか、こんな用語中学の音楽の授業以外では使われないと思っていたのだが、変更できる。この音階として7種類がサポートされていて、楽しく遊べる。

◎総評

同盟、労連、全職同、動労、大騒動、大混乱、集団自決、総合的評価を最後にしておこう。何せ、作ってるのがビクター音産である。そこここにプロの香りがする。苦勞してきた人々のウンチクがノウハウが、プログラマの苦勞が悩みがノイローゼが胃下垂が、奥にはのかに秘められている。じゃこれが実用に使えるのかミュージシャンがステージで使えるのかこれでレコードが作れるかオーケストラバックにコンチェルトでもできるか、え、えええ、どうなんているかと凄まじくは困る。そいつあ、とんだ早合点だ。とにかくおもしろいのだ。そこはMSX、遊ぶためにある機械なのだから完全に遊ばなければいけない。そのための笑わせるような則りかなプログラムがこれなのであるよ。¥5,800でこれだけ遊べれば、お得でっせダンナ。



## MSX

ミュージック  
カセット  
管理プログラム

工藤 聖 治

## 説明編

このプログラムはミュージックカセットを管理するためのもので、データの入力・追加と、カセット別・A・B面別、歌手名別、曲名別の検索が可能です。

入力、検索ともに、画面の指示に従ってください。ただし、入力の際にはデータを一時的にテープにSAVEし、最後にメインプログラムとMERGEしますので、そのためのテープを用意して録音状態にしてください。

## ●データの形式は以下のとおりです

カセット番号については、2桁のカセット番号（自分で適当に決めておく）とA・B面の別、それに何曲目かを示す数字から成り、たとえばA1B08ならば、A1カセットB面の8曲目という意味になります。

曲名、歌手名については、そのまま空白を含めての入力が可能です。

入力されたデータは、10000行から始まるDATA文としてプログラムの終わりに追加されるので、次回の追加に備えて、RENUMをパラメータなしで実行してからプログラム

全体をSAVEしておきます。SAVE検索では、

F1 キーで入力されている全データ

F2 キーでカセット別面別データ

（カセット番号+AかBの3文字を入力）

F3 キーで歌手別データ

F5 キーで曲名別データ

となっており、終了はF4キーです。

## LIST入力上の注意

LISTには工藤君自身が実際に使っている長大なデータがついていましたが、編集部ではサンプルデータとして、10個のものを用意してみました。このデータを使って検索の要領をつかんでください。そして、実際に自分のデータを入力する際には、1180行から1270行までを削除し、1160行をDATA0とします。その後は、プログラム説明のように、入力、RENUM、プログラムSAVEを繰り返すだけで、自分だけのデータベースができて上がります。

なお、オリジナルのプログラムでのデータ

入力中に文字データ領域が少なくなったので、多少変更を加えています。

投稿ソフト 評価  
StudioB.C.

これは静岡県清水市の高校生、工藤君から送られて来たものです。

SonyのMSXにはHITBIT ノートという簡易データベースが内蔵されているものもあり、ちょっとした住所録ぐらいは簡単に作れてしまうわけです。しかし考えてみれば、大辞典に対して机上辞典があるように、目的を限って気軽に使おうなんていうときには、汎用性の高いものよりもこうした専用ソフトの方が使い勝手が良いという人もいるでしょう。BASICの分かる人であれば、質問項目を変更して自分の目的（たとえば、カセットベースのオリジナルプログラムの管理など）に合わせてカスタマイズも可能。オリジナルカセットを作ったあとの管理ばかりでなく、作成するときにも、市販アルバムのデータを入力して置き、編集作業の助手としても使えるような実用ソフトでした。

```

10 CLS:SCREEN 1:COLOR 15,4,7
20 INPUT "カセット - 1? カンク - 2":T
30 IF T=1 THEN 40 ELSE 20
40 T=T: THEN 50 ELSE 30
50 -----カセット-----
60 SCREEN 0:CLS
70 INPUT "カセットのロケーション ニシテ クラウド"
80 INPUT "カセットの番号":M
90 DIM A$(M),B$(M),C$(M)
100 OPEN "CAS:CAT" FOR OUTPUT AS#1
110 FOR I=1 TO M
120 INPUT "カセットの番号":A$(I)
130 INPUT "カセットの曲名":B$(I)
140 INPUT "カセットの歌手名":C$(I)
150 NEXT I:CLS
160 SCREEN 0
170 WIDTH=40
180 FOR I=1 TO M
190 PRINT A$(I);TAB(6)B$(I);TAB(25)C$(I)
200 NEXT I
210 -----ファイル-----
220 KEY=10000
230 RESTORE 1160
240 READ JJ
250 JJ=JJ+M
260 QQ=1160:QA$=STR$(QQ)+"DATA"
270 PRINT #1,QA$:JJ
280 FOR I=1 TO M
290 B$(I)=" "+B$(I)
300 C$(I)=" "+C$(I)
310 LN$=STR$(K)+"DATA"
320 PRINT #1,LN$:A$(I);B$(I);C$(I);CHR$(8)
330 K=K+10
340 NEXT I
350 CLOSE
360 PRINT "カセットのリストをシテ スクリーンに ライブ"
370 K$=INKEY$:IF K$=" " THEN 380 ELSE 370
380 MERGE "CAS:CAT"
390 -----カンク-----
400 SCREEN 0:WIDTH=39:KEY OFF
410 RESTORE 1160
420 READ JJ
430 N=JJ
440 DIM A$(N),B$(N),C$(N),D$(N)
450 CLS:RESTORE 1180
460 FOR I=1 TO N
470 READ A$(I)
480 READ B$(I)
490 READ C$(I)
500 NEXT I
510 COLOR 7,1,1
520 CLS
530 PRINT "F1 ----- センリスト F2 ----- カセットリスト F3 ----- カシュリスト F4 -----
540 F5 ----- フォクリスト":GOTO 550
550 PRINT:PRINT "カンク カンク シマシタ
560 F3- カシュリスト F4- シュクリスト F5- フォクリスト"
570 ON KEY GOSUB 580,620,800,1140,930
580 FOR I=1 TO 5:KEY(I) ON:NEXT I
590 GOTO 550
600 CLS
610 FOR I=1 TO N
620 PRINT A$(I);TAB(7)B$(I);TAB(25)C$(I):FOR J=1 TO 75:NEXT J,I
630 GOTO 540
640 CLS
650 -----カセットリスト-----
660 CLS:INPUT "カセットの番号":Q$
670 IF LEN(Q$)<3 THEN 640
680 FOR I=1 TO N
690 Q=ASC(Q$):A=ASC(A$(I))
700 IF Q=A THEN 700
710 GOTO 770
720 P=ASC(MID$(Q$,2,1)):P=ASC(MID$(A$(I),2,1))
730 IF Y=P THEN 730
740 GOTO 770
750 U=ASC(MID$(Q$,3,1)):U=ASC(MID$(A$(I),3,1))
760 IF U=V THEN 760
770 GOTO 770
780 PRINT A$(I);TAB(7)B$(I);TAB(25)C$(I)
790 NEXT I
800 GOTO 540
810 -----カシュリスト-----
820 CLS:INPUT "カシュリスト":U$
830 IF LEN(U$)=0 THEN 800
840 Q=LEN(U$)
850 FOR I=1 TO N
860 Q=LEN(C$(I))
870 IF Q=0 THEN 860 ELSE 900
880 B=ASC(MID$(U$,3,1)):W=ASC(MID$(C$(I),3,1))
890 IF B=W THEN 880 ELSE 900
900 D=ASC(RIGHT$(U$,3)):D=ASC(RIGHT$(C$(I),3))
910 IF D=0 THEN 910 ELSE 900
920 NEXT I:GOTO 540
930 PRINT A$(I);TAB(7)B$(I);TAB(25)C$(I):GOTO 900
940 -----フォクリスト-----
950 CLS:INPUT "フォクリスト":L$
960 IF LEN(L$)=0 THEN 930
970 FOR I=1 TO L

```

```

960 INPUT "フォクリスト":D$(I)
970 IF LEN(D$(I))=0 THEN 960
980 NEXT I
990 FOR I=1 TO L
1000 G=LEN(D$(I))
1010 FOR J=1 TO N
1020 U=LEN(B$(J))
1030 IF G=U THEN 1070 ELSE 1040
1040 NEXT J
1050 NEXT I
1060 GOTO 540
1070 F=ASC(MID$(D$(I),3,1)):W=ASC(MID$(B$(J),3,1))
1080 IF F=W THEN 1070 ELSE 1040
1090 T=ASC(RIGHT$(D$(I),3)):B=ASC(RIGHT$(B$(J),3))
1100 IF T=B THEN 1120 ELSE 1040
1110 /
1120 PRINT:PRINT A$(J);TAB(7)B$(J);TAB(25)C$(J)
1130 GOTO 1040
1140 END
1150 /-----DATA-----
1160 DATA 10
1170 /
1180 DATAA1A01,Private Eyes,Hall & Oates
1190 DATAA1A02,Escape,Journey
1200 DATAA1A03,Soft Stuff,Earl Klugh
1210 DATAA1A04,All Right,C.Cross
1220 DATAA1A05,Mano a Mano,Hall & Oates
1230 DATAA1B01,Open Arms,Journey
1240 DATAA1B02,Soul Shadow,Crusaders
1250 DATAA1B03,Twinkle,Earl Klugh
1260 DATAA1B04,Arthur's Theme,C.Cross
1270 DATAA1B05,Some Men,Hall & Oates

```

## 表・変数

M	入力するデータの数No.1
A\$(I)	入力データ (番号)
B\$(I)	入力データ (曲名)
C\$(I)	入力データ (歌手名)
JJ	入力するデータの数No.2
QA\$	曲数データ
LN\$	入力データ
K	入力データの行番号
N	検索データ数
DS(I)	検索曲名
Q\$	検索番号 (カセット)
Q	Q\$の最初の文字のキャラクタ番号
A	A\$の最初の文字のキャラクタ番号
Y	Q\$の2番目の文字のキャラクタ番号
P	A\$の2番目の文字のキャラクタ番号
U	Q\$の3番目の文字のキャラクタ番号
V	A\$の3番目の文字のキャラクタ番号
U\$	検索 (歌手名)
S	U\$の文字数
D	C\$の文字数
E	U\$の3番目の文字のキャラクタ番号
W	C\$の3番目の文字のキャラクタ番号
Z	U\$の後から3番目の文字のキャラクタ番号
O	C\$の後から3番目の文字のキャラクタ番号
L	検索の曲数
G	D\$の文字数
U	B\$の文字数
F	D\$の最初から3桁目のキャラクタ番号
W	B\$の最初から3桁目のキャラクタ番号
T	D\$の最後から3桁目のキャラクタ番号
B	B\$の最後から3桁目のキャラクタ番号



# Eddy 2による 「バンゲリングベイ」予告編

## バンゲリングベイをめぐる EDDY II の反乱

ご存知のように Sony では、この夏 MSX 向け超強力ソフト、「バンゲリングベイ」(米国名 "RAID ON BUNGELING BAY" — バンゲリング湾への奇襲 —) の発売を予定しているが、これをめぐってグラフィックソフトの雄 "Graphic Ball + EDDY II" の反乱が発覚した模様である。

「バンゲリングベイ」はあの米国ブローダーバンドが開発した、「チョップリフター」「ロードランナー」に続く、俗にバンゲリング帝国ものと呼ばれるシリーズの最新作だが、発売に先がけて、そのオープニングタイトル面を盗み見た EDDY II が、得意のラフスケッチ機能、修正機能、きわめつけの BASIC プログラムジェネレート機能を生かして、Oh! HiTBiT 読者の CRT へと画像転送を試みようとしているらしい。

前号では最新の HiTBiT マシンである HB-701 と、そのバンドリングソフトである HiTBiT アートを紹介したが、既に HB-55, 75, MEZZO などをもっている読者(それにその君、野菜まがいや、怪物まがいの機種だからといって、コソコソ読んでることはないゾ!) には「だからどうした」なのであった。EDDY II はそんな読者の窮状を見るにつけ、いたたまれなくなって立ち上がったのである。

この計画の青写真によると、立案からリストまですべて EDDY II と Graphic Ball の共謀によるもので、せっかくのスクープを写真などのペーパーメディアでは何のためのバ

ソコンか? グラフ用紙のドットをひろっての CG ではどこが省力化なのか? と首をひねり、ご覧のとりのプログラムリストとなったワケなのさ。わかるかい、ハニー。さて、Oh! HiTBiT Vol. 3 の紹介では「他に修正用メニューと保存用メニューもあります。」のひとりで片付けられている EDDY II だが、実はこの修正用メニューと保存用メニューに彼の真骨頂があったわけで、これを周囲の人間が理解してやらなかったことも、今回の EDDY II の取った行動の動機のひとつではないかと思われる。

EDDY II は兄の HiTBiT アートとは絵のデータの持ち方が違うので、絵が完成した後も中途の段階までさかのぼって形・色・大きさ等の修正ができてしまうのである。しかもやはりマイコンを使って 1 ステップずつ描かせながら目で確認しつつ修正できるので、スクープしたタイトル画面など、うろ覚えであるにもかかわらずそれらしく仕上がっているのは、こうした能力のおかげだと親しいプリンタなどにもらしていたという。ちなみにこのプリンタは、以前にも図中に文字のタイプも可能な EDDY II + Graphic Ball のコンビによる絵日記(小学 6 年生男子生徒も関わっていると見られている)および HR — ホームルーム — 日誌(同じく中学校女子生徒を内儀中)の印刷を担当したことがあり、最近 EDDY II が別のカラープリンタと付き合い始めたために、今回の情報提供となったものだ。

HiTBiT アートではディスクの使用を前提とした V-RAM の一括セーブ方式を用いて画像の保存を行っていた。しかし、MSX ユーザーにはいまだにテープベースでのプログラミングを行っている者もあり、自分のプログラム中での背景の作成に頭をいためているはずなのだが、彼らの大部分が当局の追求を避けて地下に潜伏している現状では編集部も手を出せないというのが正直なところだった。ところが今回の事件で、エディタで描くだけでそのまま BASIC プログラムのジェネレートがテープに対して可能となることが明るみに出たところから、これらの潜伏組の一斉摘発も夢ではないと特捜部編集一課の N 氏は見ている。

### 手配書

氏名 Graphic Ball + EDDY II

特徴 マウスと並ぶポインティングデバイス、Graphic Ball と ROM カートリッジ化されたグラフィックスエディタのコンビで、流行のメニューアイコン表示や、ラバーバンドカーソルによる位置指定などを身につけている。又、画像の保存は、V-RAM、手順、BASIC プログラムの 3 通りを、テープおよびディスクに対して行え、プリントアウトも可能。

### 逮捕時の注意

ジェネレートされた BASIC プログラムを単体で使うときは、描線の太さの説定などを決める EDDY II 独自の拡張命令を含んでいないことを確かめること(ちなみにサンプルリストは OK である)。

0000 ///  
 0010 / RAID ON BUNGELING BAY /  
 0020 / Sneak Preview Title /  
 0030 / Drawn with EDDY 2 /  
 0040 / by Studio B.C. /  
 0050 ///////////////////////////////////  
 0060 COLOR 15,4,4;SCREEN 2  
 0070 COLOR 3  
 0080 CIRCLE (128,72),47,...,47/23  
 0090 PAINT (127,72)  
 0100 LINE (120,16)-(136,32),.BF  
 0110 LINE (80,96)-(176,104),.BF  
 0120 LINE (176,88)-(184,112),.BF  
 0130 LINE (72,88)-(80,112),.BF  
 0140 LINE (64,112)-(88,120),.BF  
 0150 LINE (168,112)-(192,120),.BF  
 0160 COLOR 4  
 0170 PSET (120,96)  
 0180 LINE (104,56)  
 0190 LINE (112,40)  
 0200 LINE (144,40)  
 0210 LINE (152,56)  
 0220 LINE (136,96)  
 0230 LINE (120,96)  
 0240 PAINT (121,76)  
 0250 COLOR 14  
 0260 PSET (120,48)  
 0270 LINE (112,72)  
 0280 LINE (144,72)  
 0290 LINE (136,48)  
 0300 LINE (120,48)  
 0310 LINE (112,40)  
 0320 LINE (144,40)  
 0330 LINE (136,48)  
 0340 LINE (120,0)-(136,4),.BF  
 0350 LINE (124,4)-(132,16),.BF  
 0360 PSET (120,4)  
 0370 LINE (124,16)  
 0380 PSET (132,16)  
 0390 LINE (136,4)  
 0400 COLOR 9  
 0410 LINE (64,120)-(88,124),.BF  
 0420 LINE (168,120)-(192,124),.BF  
 0430 COLOR 1  
 0440 LINE (68,120)-(72,124),.BF  
 0450 LINE (76,120)-(80,124),.BF  
 0460 LINE (84,120)-(88,124),.BF  
 0470 LINE (168,120)-(172,124),.BF  
 0480 LINE (176,120)-(180,124),.BF  
 0490 LINE (184,120)-(188,124),.BF  
 0500 COLOR 4  
 0510 PSET (88,112)  
 0520 LINE (88,124)  
 0530 COLOR 9  
 0540 CIRCLE (96,112),8  
 0550 CIRCLE (160,112),8  
 0560 PAINT (96,112)  
 0570 PAINT (157,113)  
 0580 COLOR 4  
 0590 CIRCLE (96,112),4  
 0600 CIRCLE (160,112),4  
 0610 PSET (147,105)  
 0620 LINE (150,95)  
 0630 PSET (136,120)  
 0640 LINE (136,96)  
 0650 COLOR 3  
 0660 PSET (110,71)  
 0670 PSET (111,71)  
 0680 PSET (111,72)  
 0690 PSET (110,72)  
 0700 COLOR 14  
 0710 PSET (113,71)  
 0720 LINE (143,71)  
 0730 COLOR 4  
 0740 LINE (147,71)-(152,72),.BF  
 0750 PSET (145,73)  
 0760 PAINT (150,68)  
 0770 PAINT (147,83)  
 0780 COLOR 12  
 0790 PSET (136,118)  
 0800 LINE (136,97)  
 0810 LINE (152,60)  
 0820 LINE (152,83)  
 0830 LINE (150,96)  
 0840 LINE (147,104)  
 0850 LINE (136,118)  
 0860 PAINT (146,86)  
 0870 COLOR 1  
 0880 PSET (108,124)  
 0890 LINE (124,120)  
 0900 LINE (132,120)  
 0910 PSET (132,120)  
 0920 LINE (148,124)  
 0930 LINE (152,128)  
 0940 LINE (148,128)  
 0950 LINE (132,124)  
 0960 LINE (124,124)  
 0970 LINE (108,128)  
 0980 LINE (104,128)  
 0990 LINE (108,124)  
 2000 PAINT (111,124)  
 2010 COLOR 9  
 2020 PSET (100,132)  
 2030 LINE (88,148)  
 2040 LINE (92,152)  
 2050 LINE (92,148)  
 2060 LINE (104,132)  
 2070 PSET (156,132)  
 2080 LINE (168,148)

2090 LINE (164,152)  
 2100 LINE (164,148)  
 2110 LINE (152,132)  
 2120 COLOR 14  
 2130 PSET (101,131)  
 2140 LINE (103,129)  
 2150 LINE (109,129)  
 2160 LINE (105,131)  
 2170 PSET (155,131)  
 2180 LINE (153,129)  
 2190 LINE (147,129)  
 2200 LINE (151,131)  
 2210 PAINT (153,130)  
 2220 PAINT (101,133)  
 2230 PSET (184,88)  
 2240 PSET (176,88)  
 2250 PSET (80,88)  
 2260 PSET (72,88)  
 2270 PSET (64,112)  
 2280 PSET (192,112)  
 2290 COLOR 12  
 2300 PSET (184,112)  
 2310 LINE (184,104)  
 2320 LINE (192,112)  
 2330 LINE (184,112)  
 2340 PSET (64,112)  
 2350 LINE (72,104)  
 2360 LINE (72,112)  
 2370 LINE (64,112)  
 2380 PAINT (67,110)  
 2390 PAINT (187,110)  
 2400 COLOR 4  
 2410 PSET (160,108)  
 2420 PSET (164,108)  
 2430 PSET (164,112)  
 2440 PSET (164,116)  
 2450 PSET (160,116)  
 2460 PSET (156,116)  
 2470 PSET (156,112)  
 2480 PSET (156,108)  
 2490 PSET (96,108)  
 2500 PSET (92,108)  
 2510 PSET (92,112)  
 2520 PSET (92,116)  
 2530 PSET (96,116)  
 2540 PSET (100,116)  
 2550 PSET (100,112)  
 2560 PSET (100,108)  
 2570 COLOR 4  
 2580 PSET (96,112)  
 2590 PSET (160,112)  
 2600 COLOR 14  
 2610 PSET (10,4)  
 2620 LINE (128,0)  
 2630 LINE (250,4)  
 2640 LINE (250,5)  
 2650 LINE (128,1)  
 2660 LINE (10,5)  
 2670 COLOR 1  
 2680 PSET (128,5)  
 2690 LINE (136,5)  
 2700 PSET (131,6)  
 2710 LINE (131,15)  
 2720 PSET (81,104)  
 2730 LINE (109,104)  
 2740 LINE (109,103)  
 2750 LINE (81,103)  
 2760 PSET (175,104)  
 2770 LINE (148,104)  
 2780 PSET (149,103)  
 2790 LINE (175,103)  
 2800 COLOR 12  
 2810 PSET (144,103)  
 2820 PSET (144,104)  
 2830 PSET (145,104)  
 2840 PSET (145,103)  
 2850 PSET (146,104)  
 2860 PSET (146,103)  
 2870 PSET (147,103)  
 2880 COLOR 1  
 2890 LINE (120,96)-(135,100),.BF  
 2900 COLOR 11  
 2910 LINE (120,100)-(135,108),.BF  
 2920 COLOR 13  
 2930 LINE (120,112)-(135,109),.BF  
 2940 COLOR 3  
 2950 PSET (111,73)  
 2960 LINE (145,73)  
 2970 LINE (135,95)  
 2980 LINE (119,95)  
 2990 PAINT (121,87)  
 3000 PSET (144,77)  
 3010 PSET (143,79)  
 3020 PSET (136,94)  
 3030 PSET (136,95)  
 3040 COLOR 12  
 3050 PSET (111,73)  
 3060 LINE (119,95)  
 3070 COLOR 14  
 3080 PSET (143,63)  
 3090 LINE (138,47)  
 3100 LINE (143,58)  
 3110 LINE (140,46)  
 3120 LINE (143,54)  
 3130 LINE (142,43)  
 3140 PSET (112,64)  
 3150 LINE (118,47)  
 3160 LINE (112,59)  
 3170 LINE (116,45)

3180 LINE (112,52)  
 3190 LINE (114,44)  
 3200 PSET (115,42)  
 3210 LINE (141,42)  
 3220 PSET (139,44)  
 3230 LINE (117,44)  
 3240 PSET (119,46)  
 3250 LINE (137,46)  
 3260 PSET (122,50)  
 3270 LINE (116,69)  
 3280 LINE (127,69)  
 3290 LINE (120,67)  
 3300 LINE (126,66)  
 3310 LINE (121,64)  
 3320 LINE (126,63)  
 3330 LINE (121,61)  
 3340 LINE (126,60)  
 3350 LINE (122,58)  
 3360 LINE (126,57)  
 3370 LINE (123,57)  
 3380 LINE (126,55)  
 3390 LINE (123,54)  
 3400 LINE (127,51)  
 3410 LINE (122,50)  
 3420 PAINT (121,60)  
 3430 COLOR 1  
 3440 PSET (80,89)  
 3450 LINE (80,95)  
 3460 PSET (79,95)  
 3470 LINE (79,88)  
 3480 PSET (80,103)  
 3490 LINE (80,111)  
 3500 PSET (79,111)  
 3510 LINE (79,103)  
 3520 PSET (134,89)  
 3530 LINE (184,103)  
 3540 PSET (183,88)  
 3550 LINE (183,112)  
 3560 LINE (85,121)-(86,123),.B  
 3570 LINE (190,121)-(191,123),.B  
 3580 PSET (74,121)  
 3590 LINE (74,123)  
 3600 PSET (82,121)  
 3610 LINE (82,123)  
 3620 PSET (174,121)  
 3630 LINE (174,123)  
 3640 PSET (182,121)  
 3650 LINE (182,123)  
 3660 COLOR 9  
 3670 LINE (69,121)-(71,123),.B  
 3680 LINE (77,121)-(79,123),.B  
 3690 LINE (85,121)-(86,123),.B  
 3700 LINE (169,121)-(171,123),.B  
 3710 LINE (177,121)-(179,123),.B  
 3720 LINE (185,121)-(187,123),.B  
 3730 COLOR 12  
 3740 LINE (79,102)-(107,101),.B  
 3750 LINE (149,102)-(175,101),.B  
 3760 LINE (120,76)-(136,80),.BF  
 3770 PSET (136,80)  
 3780 LINE (132,92)  
 3790 PSET (120,80)  
 3800 LINE (124,92)  
 3810 COLOR 1  
 3820 PSET (139,114)  
 3830 LINE (116,114)  
 3840 COLOR 1  
 3850 PSET (118,116)  
 3860 LINE (138,116)  
 3870 COLOR 1  
 3880 PSET (136,118)  
 3890 LINE (120,118)  
 3900 COLOR 12  
 3910 PSET (135,112)  
 3920 PSET (135,111)  
 3930 PSET (120,112)  
 3940 PSET (120,111)  
 3950 COLOR 1  
 3960 PSET (136,17)  
 3970 LINE (136,26)  
 3980 PSET (134,26)  
 3990 LINE (134,17)  
 4000 PSET (132,17)  
 4010 LINE (132,27)  
 4020 COLOR 9  
 4030 PSET (88,108)  
 4040 LINE (88,105)  
 4050 PSET (90,106)  
 4060 LINE (90,105)  
 4070 PSET (102,106)  
 4080 LINE (102,105)  
 4090 PSET (154,106)  
 4100 LINE (154,105)  
 4110 PSET (166,106)  
 4120 LINE (166,105)  
 4130 PSET (168,108)  
 4140 LINE (168,105)  
 4150 COLOR 1  
 4160 CIRCLE (152,56),5  
 4170 PAINT (151,56)  
 4180 COLOR 4  
 4190 CIRCLE (152,56),3  
 4200 PSET (152,56)  
 4210 COLOR 1  
 4220 PSET (120,76)  
 4230 LINE (136,76)  
 4240 CIRCLE (128,128),8  
 4250 CIRCLE (128,136),4  
 4260 PSET (121,132)

4270 LINE (124,137)  
 4280 PSET (135,132)  
 4290 LINE (132,137)  
 4300 PAINT (129,128)  
 4310 COLOR 14  
 4320 PSET (128,129)  
 4330 LINE (128,125)  
 4340 PSET (126,130)  
 4350 LINE (126,125)  
 4360 PSET (124,130)  
 4370 LINE (124,126)  
 4380 COLOR 4  
 4390 PSET (126,136)  
 4400 LINE (130,136)  
 4410 PSET (125,135)  
 4420 PSET (131,135)  
 4430 COLOR 14  
 4440 PSET (125,138)  
 4450 PSET (125,139)  
 4460 PSET (127,140)  
 4470 PSET (127,137)  
 4480 COLOR 1  
 4490 PSET (128,137)  
 4500 PSET (129,137)  
 4510 PSET (130,136)  
 4520 PSET (129,135)  
 4530 PSET (128,135)  
 4540 PSET (127,135)  
 4550 PSET (126,136)  
 4560 COLOR 15  
 4570 PSET (24,32)  
 4580 LINE (36,32)  
 4590 LINE (40,36)  
 4600 LINE (40,44)  
 4610 LINE (36,48)  
 4620 LINE (40,52)  
 4630 LINE (40,64)  
 4640 LINE (36,64)  
 4650 LINE (36,56)  
 4660 LINE (32,52)  
 4670 LINE (28,52)  
 4680 LINE (28,64)  
 4690 LINE (24,64)  
 4700 LINE (24,32)  
 4710 PSET (28,36)  
 4720 LINE (28,44)  
 4730 LINE (36,44)  
 4740 LINE (36,36)  
 4750 LINE (28,36)  
 4760 PSET (48,64)  
 4770 LINE (48,36)  
 4780 LINE (52,32)  
 4790 LINE (60,32)  
 4800 LINE (64,36)  
 4810 LINE (64,64)  
 4820 LINE (60,64)  
 4830 LINE (60,52)  
 4840 LINE (52,52)  
 4850 LINE (52,64)  
 4860 LINE (48,64)  
 4870 PSET (52,44)  
 4880 LINE (60,44)  
 4890 LINE (60,36)  
 4900 LINE (52,36)  
 4910 LINE (52,44)  
 4920 LINE (72,32)-(76,64),.B  
 4930 PSET (84,32)  
 4940 LINE (84,64)  
 4950 LINE (96,64)  
 4960 LINE (100,60)  
 4970 LINE (100,36)  
 4980 LINE (96,32)  
 4990 LINE (84,32)  
 5000 LINE (88,36)-(96,60),.B  
 5010 PSET (160,36)  
 5020 LINE (160,60)  
 5030 LINE (164,64)  
 5040 LINE (172,64)  
 5050 LINE (176,60)  
 5060 LINE (176,36)  
 5070 LINE (172,32)  
 5080 LINE (164,32)  
 5090 LINE (160,36)  
 5100 LINE (164,36)-(172,60),.B  
 5110 PSET (184,32)  
 5120 LINE (184,64)  
 5130 LINE (188,64)  
 5140 LINE (188,40)  
 5150 LINE (196,64)  
 5160 LINE (200,64)  
 5170 LINE (200,32)  
 5180 LINE (196,32)  
 5190 LINE (196,56)  
 5200 LINE (188,32)  
 5210 LINE (184,32)  
 5220 PSET (24,124)  
 5230 LINE (48,124)  
 5240 LINE (56,132)  
 5250 LINE (56,148)  
 5260 LINE (48,156)  
 5270 LINE (56,164)  
 5280 LINE (56,180)  
 5290 LINE (48,188)  
 5300 LINE (24,188)  
 5310 LINE (24,124)  
 5320 PSET (32,132)  
 5330 LINE (48,132)  
 5340 LINE (48,148)  
 5350 LINE (44,152)



5360 LINE -(32,152)  
5370 LINE -(32,132)  
5380 PSET (32,160)  
5390 LINE -(44,160)  
5400 LINE -(48,164)  
5410 LINE -(48,180)  
5420 LINE -(32,180)  
5430 LINE -(32,160)  
5440 PSET (64,156)  
5450 LINE -(64,184)  
5460 LINE -(68,188)  
5470 LINE -(76,188)  
5480 LINE -(80,184)  
5490 LINE -(80,156)  
5500 LINE -(76,156)  
5510 LINE -(76,184)  
5520 LINE -(68,184)  
5530 LINE -(68,156)  
5540 LINE -(64,156)  
5550 PSET (88,156)  
5560 LINE -(88,188)  
5570 LINE -(92,188)  
5580 LINE -(92,164)  
5590 LINE -(100,188)  
5600 LINE -(104,188)  
5610 LINE -(104,156)  
5620 LINE -(100,156)  
5630 LINE -(100,180)  
5640 LINE -(92,156)  
5650 LINE -(88,156)  
5660 PSET (112,160)  
5670 LINE -(116,156)  
5680 LINE -(124,156)  
5690 LINE -(128,160)  
5700 LINE -(128,164)  
5710 LINE -(124,164)  
5720 LINE -(124,160)  
5730 LINE -(116,160)  
5740 LINE -(116,184)  
5750 LINE -(124,184)  
5760 LINE -(124,176)  
5770 LINE -(120,176)  
5780 LINE -(120,168)  
5790 LINE -(128,168)  
5800 LINE -(128,184)  
5810 LINE -(124,188)  
5820 LINE -(116,188)  
5830 LINE -(112,184)  
5840 LINE -(112,160)  
5850 PSET (136,156)  
5860 LINE -(136,188)  
5870 LINE -(152,188)  
5880 LINE -(152,184)  
5890 LINE -(140,184)  
5900 LINE -(140,176)  
5910 LINE -(152,176)  
5920 LINE -(152,168)  
5930 LINE -(140,168)  
5940 LINE -(140,160)  
5950 LINE -(152,160)  
5960 LINE -(152,156)  
5970 LINE -(136,156)  
5980 PSET (160,156)  
5990 LINE -(160,188)  
6000 LINE -(176,188)  
6010 LINE -(176,184)  
6020 LINE -(164,184)  
6030 LINE -(164,156)  
6040 LINE -(160,156)  
6050 LINE -(184,156)-(188,188),,B  
6060 PSET (196,156)  
6070 LINE -(196,188)  
6080 LINE -(200,188)  
6090 LINE -(200,164)  
6100 LINE -(208,188)  
6110 LINE -(212,188)  
6120 LINE -(212,156)  
6130 LINE -(208,156)  
6140 LINE -(208,180)  
6150 LINE -(200,156)  
6160 LINE -(196,156)  
6170 PSET (224,156)  
6180 LINE -(232,156)  
6190 LINE -(236,160)  
6200 LINE -(236,164)  
6210 LINE -(232,164)  
6220 LINE -(232,160)  
6230 LINE -(224,160)  
6240 LINE -(224,184)  
6250 LINE -(232,184)  
6260 LINE -(232,176)  
6270 LINE -(228,176)  
6280 LINE -(228,168)  
6290 LINE -(236,168)  
6300 LINE -(236,184)  
6310 LINE -(232,188)  
6320 LINE -(224,188)  
6330 LINE -(220,184)  
6340 LINE -(220,160)  
6350 LINE -(224,156)  
6360 PAINT (28,48)  
6370 PAINT (55,48)  
6380 PAINT (73,47)  
6390 PAINT (85,47)  
6400 PAINT (161,50)  
6410 PAINT (185,49)  
6420 PAINT (26,156)  
6430 PAINT (65,174)  
6440 PAINT (89,172),

6450 PAINT (114,170)  
6460 PAINT (140,170)  
6470 PAINT (162,168)  
6480 PAINT (186,171)  
6490 PAINT (198,171)  
6500 PAINT (222,168)  
6510 COLOR 12  
6520 LINE (81,105)-(87,111),,BF  
6530 LINE (169,105)-(175,111),,BF  
6540 COLOR 15  
6550 PSET (172,156)  
6560 LINE -(172,125)  
6570 LINE -(185,125)  
6580 LINE -(188,128)  
6590 LINE -(188,136)  
6600 LINE -(184,140)  
6610 LINE -(188,144)  
6620 LINE -(188,152)  
6630 LINE -(184,156)  
6640 LINE -(172,156)  
6650 LINE (176,128)-(184,136),,B  
6660 LINE (176,144)-(184,152),,B  
6670 PSET (196,156)  
6680 LINE -(196,128)  
6690 LINE -(200,124)  
6700 LINE -(208,124)  
6710 LINE -(212,128)  
6720 LINE -(212,156)  
6730 LINE -(208,156)  
6740 LINE -(208,144)  
6750 LINE -(200,144)  
6760 LINE -(200,156)  
6770 PSET (200,128)  
6780 LINE -(208,128)  
6790 LINE -(208,136)  
6800 LINE -(200,136)  
6810 LINE -(200,128)  
6820 PSET (220,124)  
6830 LINE -(220,144)  
6840 LINE -(232,144)  
6850 PSET (232,144)  
6860 LINE -(232,152)  
6870 LINE -(220,152)  
6880 LINE -(220,156)  
6890 LINE -(232,156)  
6900 LINE -(236,152)  
6910 LINE -(236,124)  
6920 LINE -(232,124)  
6930 LINE -(232,136)  
6940 LINE -(224,136)  
6950 LINE -(224,124)  
6960 LINE -(220,124)  
6970 PAINT (175,140)  
6980 PAINT (204,138)  
6990 PAINT (226,137)  
7000 COLOR 4  
7010 PSET (168,148)  
7020 PSET (88,148)  
7030 PSET (171,156)  
7040 LINE -(233,156)  
7050 COLOR 1  
7060 PSET (102,130)  
7070 LINE -(108,130)  
7080 PSET (100,132)  
7090 LINE -(104,132)  
7100 PSET (148,130)  
7110 LINE -(154,130)  
7120 PSET (152,132)  
7130 LINE -(156,132)  
7140 COLOR 9  
7150 LINE (24,44)-(36,52),,BF  
7160 LINE (48,44)-(64,52),,BF  
7170 LINE (72,44)-(76,52),,BF  
7180 LINE (84,44)-(88,52),,BF  
7190 LINE (96,44)-(100,52),,BF  
7200 LINE (160,44)-(164,52),,BF  
7210 LINE (172,44)-(176,52),,BF  
7220 LINE (184,44)-(188,52),,BF  
7230 LINE (196,44)-(200,52),,BF  
7240 LINE (24,168)-(32,176),,BF  
7250 LINE (48,168)-(56,176),,BF  
7260 LINE (64,168)-(68,176),,BF  
7270 LINE (76,168)-(80,176),,BF  
7280 LINE (88,168)-(92,176),,BF  
7290 LINE (100,168)-(104,176),,BF  
7300 LINE (112,168)-(116,176),,BF  
7310 LINE (120,168)-(128,176),,BF  
7320 LINE (136,168)-(152,176),,BF  
7330 LINE (160,168)-(164,176),,BF  
7340 LINE (184,168)-(188,176),,BF  
7350 LINE (196,168)-(200,176),,BF  
7360 LINE (208,168)-(212,176),,BF  
7370 LINE (220,168)-(224,176),,BF  
7380 LINE (228,168)-(236,176),,BF  
7390 LINE (172,136)-(184,144),,BF  
7400 LINE (196,136)-(212,144),,BF  
7410 LINE (220,136)-(236,144),,BF  
7420 LINE (24,136)-(32,144),,BF  
7430 LINE (48,136)-(56,144),,BF  
7440 LINE (24,42)-(28,41),,B  
7450 LINE (36,42)-(40,41),,B  
7460 PSET (37,44)  
7470 LINE -(40,44)  
7480 PSET (37,45)  
7490 LINE -(39,45)  
7500 PSET (37,51)  
7510 LINE -(39,51)  
7520 PSET (37,52)  
7530 LINE -(40,52)

7540 LINE (48,42)-(52,41),,B  
7550 LINE (60,42)-(64,41),,B  
7560 LINE (72,42)-(76,41),,B  
7570 LINE (84,42)-(88,41),,B  
7580 LINE (96,42)-(100,41),,B  
7590 LINE (160,42)-(164,41),,B  
7600 LINE (172,42)-(176,41),,B  
7610 LINE (184,42)-(191,41),,B  
7620 LINE (196,42)-(200,41),,B  
7630 PSET (189,44)  
7640 LINE -(191,44)  
7650 PSET (192,52)  
7660 LINE -(195,52)  
7670 LINE (24,54)-(28,55),,B  
7680 PSET (34,54)  
7690 LINE -(40,54)  
7700 PSET (35,55)  
7710 LINE -(40,55)  
7720 LINE (48,54)-(52,55),,B  
7730 LINE (60,54)-(64,55),,B  
7740 LINE (72,54)-(76,55),,B  
7750 LINE (84,54)-(88,55),,B  
7760 LINE (96,54)-(100,55),,B  
7770 LINE (160,54)-(164,55),,B  
7780 LINE (172,54)-(176,55),,B  
7790 LINE (184,54)-(188,55),,B  
7800 LINE (196,54)-(200,55),,B  
7810 LINE (24,134)-(32,133),,B  
7820 LINE (48,134)-(56,133),,B  
7830 LINE (72,134)-(80,133),,B  
7840 LINE (96,134)-(104,133),,B  
7850 LINE (128,134)-(136,133),,B  
7860 LINE (160,134)-(168,133),,B  
7870 LINE (192,134)-(200,133),,B  
7880 LINE (224,134)-(232,133),,B  
7890 PSET (93,168)  
7900 LINE -(95,168)  
7910 PSET (96,176)  
7920 LINE -(99,176)  
7930 LINE (100,168)-(104,165),,B  
7940 LINE (112,168)-(116,165),,B  
7950 LINE (136,168)-(140,165),,B  
7960 LINE (160,168)-(164,165),,B  
7970 COLOR 9  
7980 LINE (184,168)-(188,165),,B  
7990 LINE (196,168)-(200,165),,B  
8000 LINE (208,168)-(212,165),,B  
8010 LINE (220,168)-(224,165),,B  
8020 LINE (24,178)-(32,179),,B  
8030 LINE (48,178)-(56,179),,B  
8040 LINE (72,178)-(80,179),,B  
8050 LINE (96,178)-(104,179),,B  
8060 LINE (128,178)-(136,179),,B  
8070 LINE (160,178)-(164,179),,B  
8080 LINE (192,178)-(196,179),,B  
8090 LINE (224,178)-(228,179),,B  
8100 LINE (24,178)-(32,179),,B  
8110 LINE (48,178)-(56,179),,B  
8120 LINE (72,178)-(80,179),,B  
8130 LINE (96,178)-(104,179),,B  
8140 LINE (128,178)-(136,179),,B  
8150 LINE (160,178)-(164,179),,B  
8160 LINE (192,178)-(196,179),,B  
8170 LINE (224,178)-(228,179),,B  
8180 LINE (24,134)-(32,133),,B  
8190 LINE (48,134)-(56,133),,B  
8200 LINE (72,134)-(80,133),,B  
8210 LINE (96,134)-(104,133),,B  
8220 LINE (128,134)-(136,133),,B  
8230 LINE (160,134)-(164,133),,B  
8240 LINE (184,134)-(188,133),,B  
8250 LINE (208,134)-(212,133),,B  
8260 LINE (232,134)-(236,133),,B  
8270 LINE (24,146)-(32,147),,B  
8280 LINE (48,146)-(56,147),,B  
8290 LINE (72,146)-(80,147),,B  
8300 LINE (96,146)-(104,147),,B  
8310 LINE (128,146)-(136,147),,B  
8320 LINE (160,146)-(164,147),,B  
8330 LINE (192,146)-(196,147),,B  
8340 LINE (224,146)-(228,147),,B  
8350 LINE (24,39)  
8360 PSET (24,39)  
8370 LINE -(28,39)  
8380 PSET (36,39)  
8390 LINE -(40,39)  
8400 PSET (48,39)  
8410 LINE -(52,39)  
8420 PSET (60,39)  
8430 LINE -(64,39)  
8440 PSET (72,39)  
8450 LINE -(76,39)  
8460 PSET (84,39)  
8470 LINE -(88,39)  
8480 PSET (96,39)  
8490 LINE -(100,39)  
8500 PSET (160,39)  
8510 LINE -(164,39)  
8520 PSET (172,39)  
8530 LINE -(176,39)  
8540 PSET (184,39)  
8550 LINE -(188,39)  
8560 PSET (196,39)  
8570 LINE -(200,39)  
8580 PSET (24,57)  
8590 LINE -(28,57)  
8600 PSET (36,57)  
8610 LINE -(40,57)  
8620 PSET (48,57)  
8630 LINE -(52,57)  
8640 PSET (60,57)  
8650 LINE -(64,57)  
8660 PSET (72,57)  
8670 LINE -(76,57)  
8680 PSET (84,57)  
8690 LINE -(88,57)  
8700 PSET (96,57)

8630 LINE -(100,57)  
8640 PSET (160,57)  
8650 LINE -(164,57)  
8660 PSET (172,57)  
8670 LINE -(176,57)  
8680 PSET (184,57)  
8690 LINE -(188,57)  
8700 PSET (194,57)  
8710 LINE -(200,57)  
8720 PSET (24,131)  
8730 LINE -(55,131)  
8740 PSET (24,149)  
8750 LINE -(32,149)  
8760 PSET (47,149)  
8770 LINE -(55,149)  
8780 PSET (24,163)  
8790 LINE -(32,163)  
8800 PSET (47,163)  
8810 LINE -(55,163)  
8820 PSET (64,163)  
8830 LINE -(68,163)  
8840 LINE (76,165)-(80,166),,B  
8850 PSET (76,163)  
8860 LINE -(80,163)  
8870 PSET (88,163)  
8880 LINE -(94,163)  
8890 PSET (100,163)  
8900 LINE -(104,163)  
8910 PSET (112,163)  
8920 LINE -(116,163)  
8930 PSET (124,163)  
8940 LINE -(128,163)  
8950 PSET (136,163)  
8960 LINE -(140,163)  
8970 PSET (160,163)  
8980 LINE -(164,163)  
8990 PSET (184,163)  
9000 LINE -(188,163)  
9010 PSET (196,163)  
9020 LINE -(202,163)  
9030 PSET (208,163)  
9040 LINE -(212,163)  
9050 PSET (220,163)  
9060 LINE -(224,163)  
9070 PSET (232,163)  
9080 LINE -(236,163)  
9090 PSET (24,181)  
9100 LINE -(55,181)  
9110 PSET (64,181)  
9120 LINE -(68,181)  
9130 PSET (76,181)  
9140 LINE -(80,181)  
9150 PSET (88,181)  
9160 LINE -(92,181)  
9170 PSET (98,181)  
9180 LINE -(104,181)  
9190 PSET (112,181)  
9200 LINE -(116,181)  
9210 PSET (124,181)  
9220 LINE -(128,181)  
9230 PSET (136,181)  
9240 LINE -(140,181)  
9250 PSET (160,181)  
9260 LINE -(164,181)  
9270 PSET (184,181)  
9280 LINE -(188,181)  
9290 PSET (196,181)  
9300 LINE -(200,181)  
9310 PSET (206,181)  
9320 LINE -(212,181)  
9330 PSET (220,181)  
9340 LINE -(224,181)  
9350 PSET (232,181)  
9360 LINE -(236,181)  
9370 PSET (172,131)  
9380 LINE -(176,131)  
9390 PSET (184,131)  
9400 LINE -(188,131)  
9410 PSET (196,131)  
9420 LINE -(200,131)  
9430 PSET (208,131)  
9440 LINE -(212,131)  
9450 PSET (220,131)  
9460 LINE -(224,131)  
9470 PSET (232,131)  
9480 LINE -(236,131)  
9490 PSET (172,149)  
9500 LINE -(176,149)  
9510 PSET (184,149)  
9520 LINE -(188,149)  
9530 PSET (196,149)  
9540 LINE -(200,149)  
9550 PSET (208,149)  
9560 LINE -(212,149)  
9570 PSET (232,149)  
9580 LINE -(236,149)  
9590 GOTO 9590





企画編集や新聞の切り出し、整理プログラムを自分でつくって使っています  
（『私とパソコン』より）  
田村 健さん  
「朝日ニュースセンター9時」  
ニュースター

## お申込みは今がチャンス。 裏面のハガキでいますぐ！

これからも見逃せない情報が登場

●6月24日号特集：米国のパソコン最新ルポ

そして

●7月15日には、恒例・好評の新製品集中特大号を発行！

今年からは特にパソコン・ショップ店頭での実勢価格も収録



「スーパーイージーなパソコンを絶対つくってみせる」  
（『編集長インタビュー』より）  
ウィリアム・H・ゲーツ氏  
マイクロソフト会長

# パソコンを手軽に使いたい……

最近好評をいただいた記事の一部をご紹介します。  
このような記事があなたのパソコン活用をサポートします。

- ワープロ専用機VS.パソコン・ワープロ徹底比較
- 世界を広げるテレ・コンピュータリング
- 市販ソフト・ベストセラーランキング
- 携帯パソコン活用術——急速に拡大する実務への応用
- パソコン活用に役立つ失敗の研究
- 個人でもつくれる国際的パソコン・ネット
- 普及にはずみつく小型ハードディスク
- 教育に変革迫るパソコン活用
- 表計算ソフト33本徹底評価
- 戦略研究：日本電気／富士通
- 身近なパソコンから送れる電子メール
- パソコン導入後の利用術を探る
- 日本語ワープロソフト30本徹底評価
- 米国で猛威振うコンピューター犯罪

■年間予約制ならではの割安な料金

★1年（17冊） 6,000円……1冊当たり353円

★3年（51冊）17,000円……1冊当たり333円

発行：日経マگزロービル社



1冊当たり  
わずか235円

「日本人による日本人のためのコンピュータが必要」  
（『編集長インタビュー』より）  
田村 健氏  
第一理研情報科学科助手

「一番利用度の高いのは、  
なんといっても原稿を書く  
ためのワープロソフト」  
（『私とパソコン』より）  
岩本久則氏  
漫画家



パソコンを仕事に使うためのわかりやすい情報誌

## 日経パソコン

NIKKEI PERSONAL COMPUTING





# 1冊当たりわずか235円!

気軽にお読みにませんか

パソコンを仕事に使うためのわかりやすい情報誌

## 日経パソコン

NIKKEI PERSONAL COMPUTING

1年17冊、毎号このような編集の柱で、雑誌づくりを進めています。

- 毎号、話題の市販ソフトを徹底評価
- パソコンの意欲的な活用事例を数多く紹介
- さまざまなパソコン関連ニュースを国内外から収集し、紹介
- パソコン基礎講座／用語辞典などで、パソコン活用方法をわかりやすく解説
- 年2回、ハード／ソフトの新製品集中特大号

【年2回の新製品集中特大号】7月に前期号を発行

半年間に発表されたハード／ソフトの新製品情報を、通常号の頁を倍増して集中収録。独自企画の調査を実施し、ほかでは得られないデータ、例えば●バージョン・アップ時やバグへの対応●ハード本体のサイズや使用電力などに加えて、今回より●パソコン・ショップでの実勢価格を掲載。製品選択に不可欠の本格的なバイヤーズ・ガイドとして、たいへん好評をいただいています。

【ソフト情報を毎号豊富に掲載】

続々と登場するソフトの中から、注目すべきものを取りあげ、ユーザーの立場から、性能・機能、マニュアルの出来、使い勝手などを厳しくチェックします。

【パソコンの活用事例を数多く紹介】

国内外を専門記者が直接取材し、先進的なパソコン活用を行っているケースを紹介。ビジネスの場はもとより、教育界、家庭などでのパソコン利用の実態を読みとれます。

【年間予約購読制ならではの割安な料金】

★1年(17冊) 6,000円…1冊当たり353円

★3年(51冊) 12,000円…1冊当たり235円

料金お支払いにつきましては別途ご案内を差しあげます。

● 毎3週刊(3週間に1回、年17冊発行)

● A4変型判、毎号約200頁

● 個人の年間予約購読、ご自宅郵送制

発行：日経マクロウヒル社

郵便はがき

168-00

料金を  
取人払

(受取人)

東京都杉並区浜田山4-5-5

杉並南郵便局私書箱35号

日経マクロウヒル社

読者サービス・センター

NPC係行

日経パソコン 購読申込書

読者のプロフィールを正確に把握し、購読の便を図るため、以下の事項を記入してください。なお、購読料は別途お支払いください。

お名前(フリガナ) 年齢 性別 職業 会社名 住所(フリガナ) 郵便番号

ご自宅住所(フリガナ) 郵便番号

購読料(1年17冊) 6,000円(1冊当たり353円) 3年(51冊) 12,000円(1冊当たり235円)

購読料をお支払い方法(いずれかに○印を) 1. 郵便局からの払い込み 2. 銀行振込(口座振替) 3. 銀行口座からの引き落とし

職種(該当のものに○印を) 1. 企業経営者 2. 商工サービス員 3. 管理職 4. 専門職 5. 技術職 6. 事務職 7. 販売職 8. サービス職 9. 学生 10. その他

出身学部(該当のものに○印を) 1. 大学 2. 短期大学 3. 専門学校 4. 高校 5. その他

お申し込みは、このハガキでいますぐ!

7月発行の新製品集中特大号もお読みになれます。

お申し込みは、このハガキでいますぐ!



# FLOPPY

MAGAZINE

Dizzy Rider・リアルタイムゲーム  
アドベンチャーゲーム・エディタ  
Car Graphic Design  
デジタル・ミュージック・神谷 嘉徳

777 777C

PRINT

Opera

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100

## フロッピーマガジンでひろがる世界

# FLOPPY

MAGAZINE

フロッピーマガジン=フロッピー+ペーパーメディア  
for SMC-777/777C



フロッピーマガジンをSMC-777/777Cにセットし、  
スイッチをONするとパソコンで楽しむマガジンになります。  
また、フロッピーマガジンには  
グラフィカルで、カラフルな構成の印刷メディアもついています。

6号発売中 ¥5,800 (ディスク2枚)  
●7号は、7月22日発売予定  
全国のソニーショップ、電気店でお求めください。

### 6号 CONTENTS

#### ディスク A ①「欧州鉄道の旅——美術館めぐり」

トーマス・クックの時刻表に基づいた、欧州の鉄道をシミュレート。7日間で多くの美術館をめぐり、各館の説明や名画をお楽しみください。本文では、欧州の国際列車—TGV オリエント急行、ラインゴルト—による旅のもつ魅力を紹介。

#### ②ジグソーパズル+マップクイズ

地図ジグソーの第3弾。欧州鉄道の旅にちなんで、テーマは「ヨーロッパ」。ジグソーパズルの他に、都市名あてクイズもあります。

#### ③ツール・ディスクユーティリティ

ディスクの空きエリアやファイルのサイズがBASIC上でわかるマシン語パッケージ。本文に使い方を解説。

#### ディスク B ①デジタル・ミュージック

運動会でおなじみの「クシコスポスト」のアレンジ。本文では、ラッサビーター実戦編曲講座その3。神谷スタジオでは「変態奏法」といわれているテクニックを公開。

#### ②English Crossword Puzzle

テーマは「夏の日英米事情」。英文解説からChallengeしてください。

#### ③パズルゲーム「ソリティア」

日本では通称「ピアス」。ヨーロッパに古くから伝わるパズルです。わからない人には、解答編もあります。本文で遊び方や解法のアルゴリズムを解説。

#### ④Ambient CG その2

都市化のスプロール現象やライフゲームのイメージを基に構成。あなたの町の推移が、見えるかもしれません。時々気がついたように眺めてください。

#### ⑤リアルタイムゲーム「Dizzy Rider」

縦、横スクロール型ゲーム。バイクのテクニックを駆使して思いがけない障害物をクリアしながら、宝石を持ち帰ってください。

#### ⑥アドベンチャーゲーム・エディタ

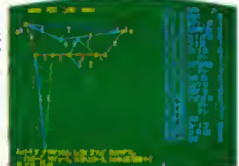
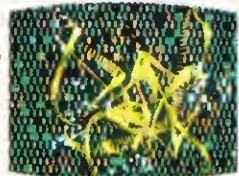
コンピュータの指示に従い入力してゆくだけで、テキスト版アドベンチャーが作れるツール。デモデータでゲームもできます。本文ではアドベンチャーゲーム・エディタのコンセプトをまとめました。

#### ⑦計画の科学「PERT」入門

プロジェクト管理のために開発された手法「PERT」の対話型入門プログラム。本文では「PERT」の概略とプログラムの使い方についてまとめました。

#### ⑧Car Graphic Design Data

グラフィック・エディターを使ってお絵描きするために、車の線画データを取りました。本文では、正確な入力が必要とされるグラフィック作成上のノウハウを公開。巻末に特別付録として画面のドット比率と対応した方眼紙をつけてあります。読者の作品を募集中です。





60年度試験受付はじまる!

7月特別号の特別付録で受験手続ができます。

コンピュータ技術者必携▷第2種・第1種・特種受験

# 月刊 情報処理試験

7月号の主な内容

特別付録

昭和60年度情報処理技術者試験案内書、  
受験願書、振替用紙、受験ガバ

特集

COBOLはこうやれば  
必ず解ける!

《特種記述問題》攻略の  
重要ポイント①

算法と流れ図の合格答案はこうつくる①

●実力がめきめきつく評判の講座

合格最短ゼミ CAP-X・FORTRAN・COBOL・PL/I

受験のための コンピュータ基礎 重点征服ゼミ 関連知識

●受験と仕事に役立つ話題の記事

90年代をみざす第5世代コンピュータ開発

合格者ルポ コンピュータが灯りをともした

大脳生理学でわかった能率的学習法

対談 人間とコンピュータ

●大好評のカラーグラフ・連載コラム・SFなど

カラーグラフ コンピュータが美女をつくる 受験なんでも相談室 ほかに



7月特別号 定価580円

好評発売中!

「月刊 情報処理試験」  
の確実な入手法

毎号、売切れ店が続出し、大変ご迷惑をおかけいたしております。「月刊情報処理試験」をお買いもらしなく確実に入手なさるには、お近くの書店で予約購読なさるのが一番確実です。お近くに書店がないなどの理由で、直接当社にご注文くださる場合には、本誌と込みの専用振替用紙をご利用ください。

お問合せ先

**SOFT  
BANK**

日本ソフトバンク 出版部

〒102 東京都千代田区四番町 2-1 ☎03(261)4095



# Beep

MAGAZINE FOR GAME KIDS

7月号

## ゲーム情報満載!

定価360yen

### 史上最大、夢のビッグスポーツイベント ◆◆◆スポーツ・エキスポ'85

ファミコン・サッカー、ハイパーオリンピックから、  
アップル・サマーゲームズまで人気スポーツゲーム大集合!  
[競技種目]サッカー●体操●ピンポン●陸上●クレイ●テニス●モトクロス  
●レスリング●アスレチック●野球●バスケット●カーレース●水泳●フットボールほか

◆徹底研究/ドラゴンスレイヤー——魔宮の戦い——

◆ビデオゲーム・ラボ

テディーボーイ・ブルース/グラディウスほか

◆ファミコン・ゲーム

スターフォース/ディグダグ/  
フラッピー/忍者くん

好評発売中!

GO  
IVAT GIL

SOFT  
BANK

日本ソフトバンク

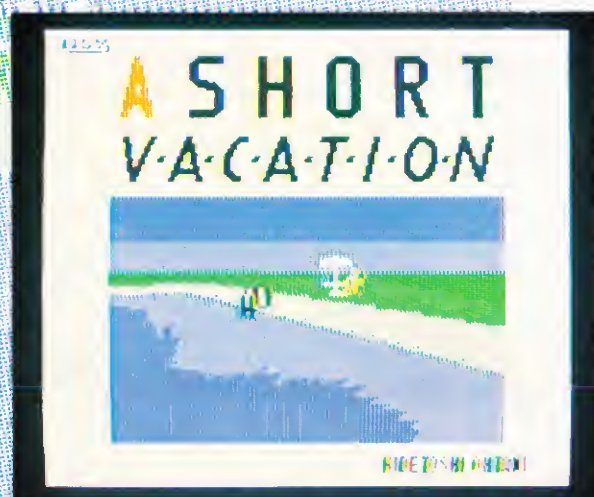


# HITBIT

## SHORT VACATION

KEY

### MSX HyoKin Land

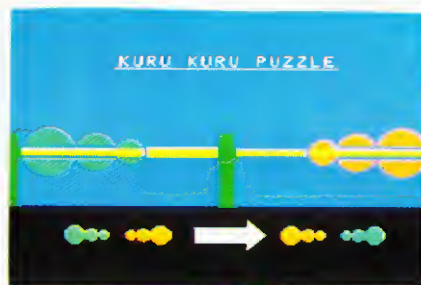
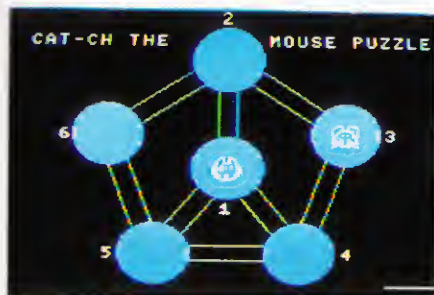


### THE MINE

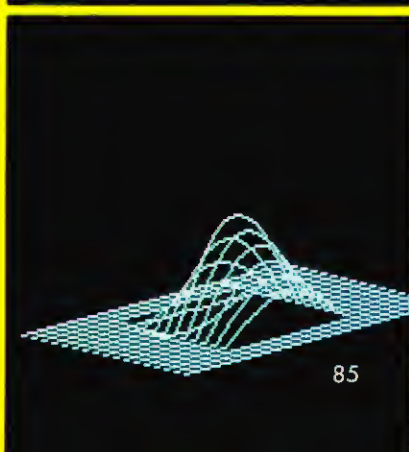
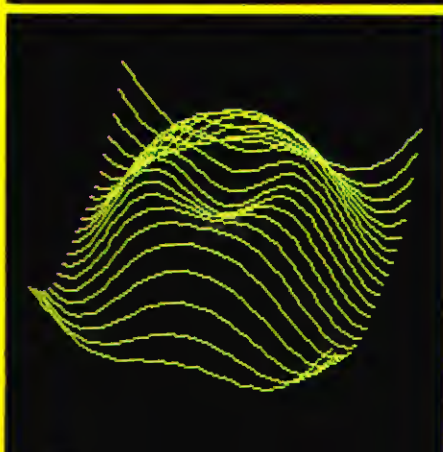
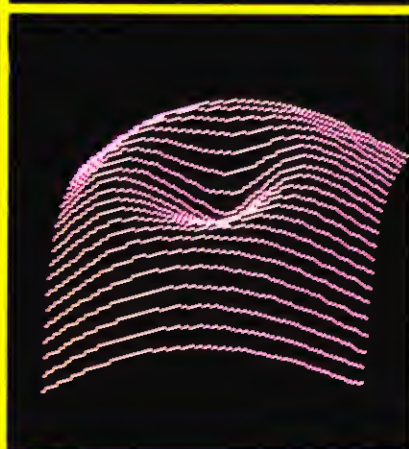
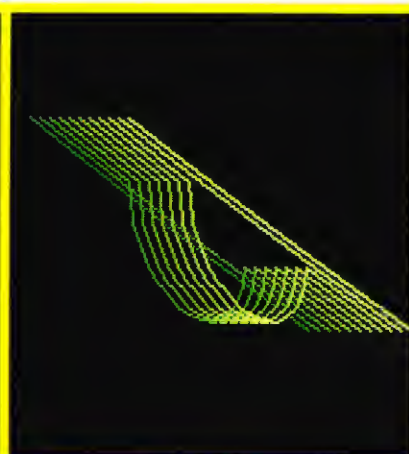




# MSX Computer Puzzler



**Pseudo  
3-D  
Graphics**



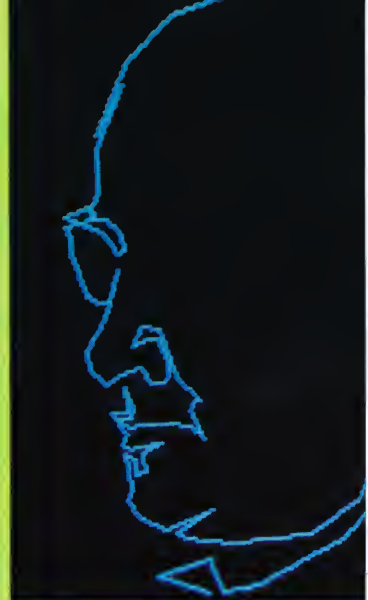




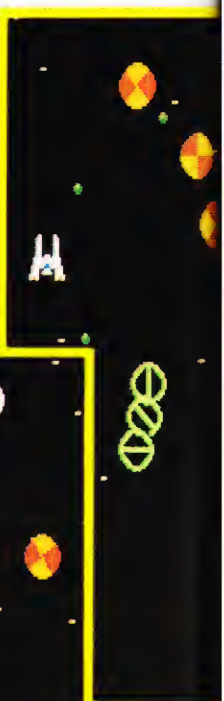
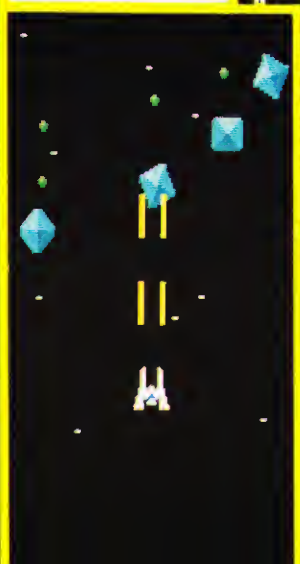
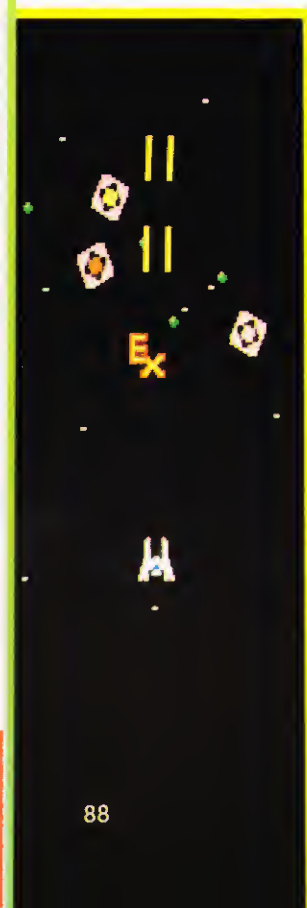
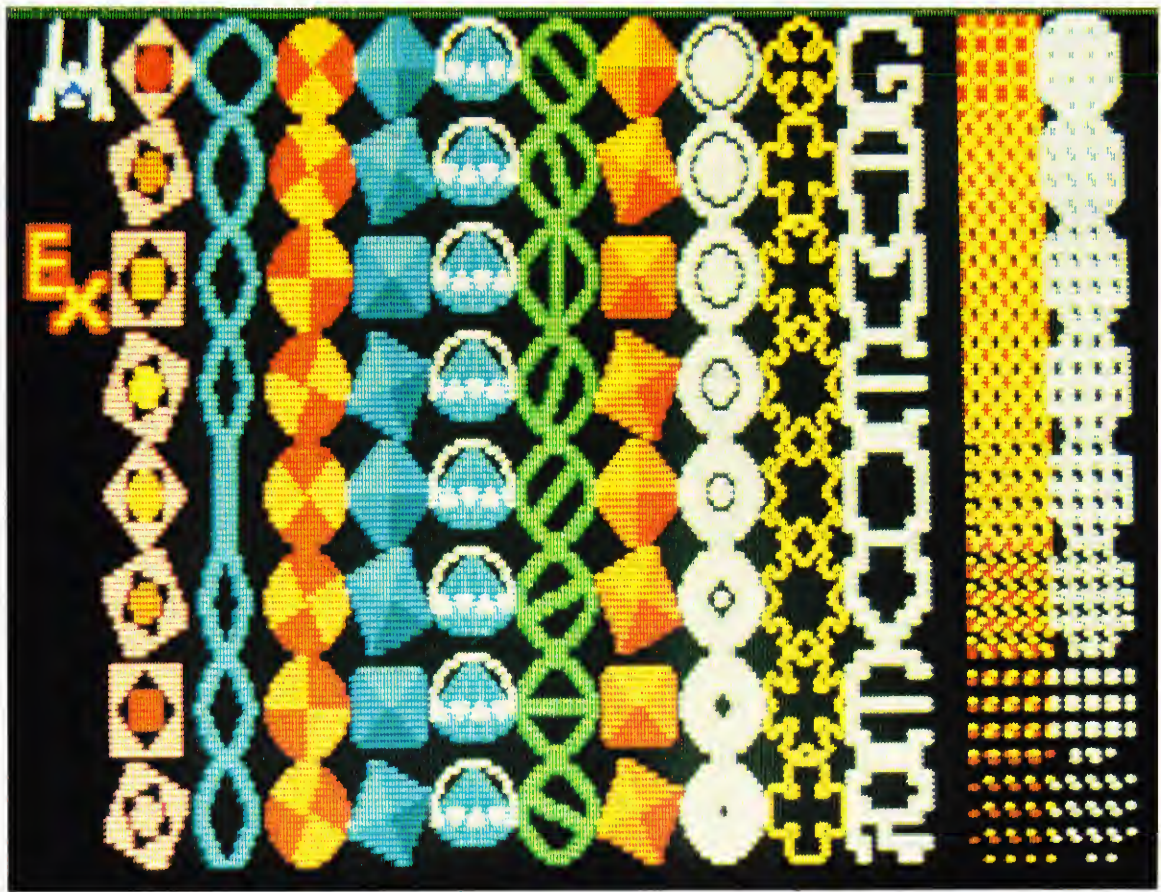
# METAMORPHOSIS



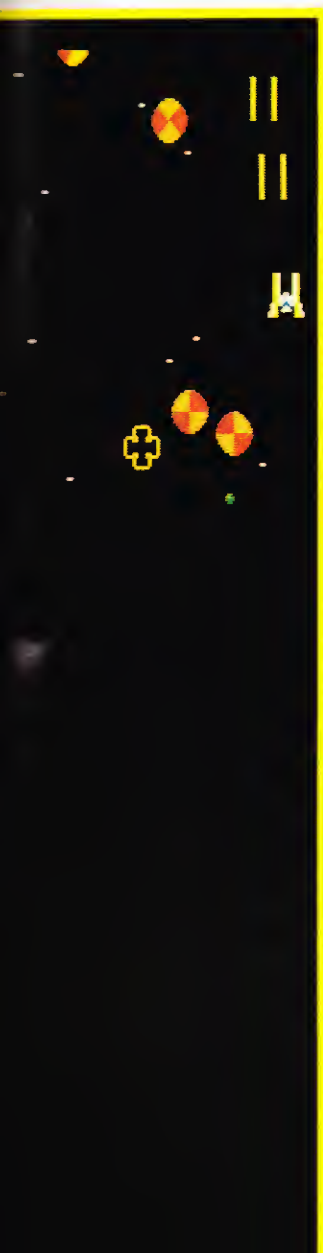
# METAMORPHOSIS







# AREA-X





## うちのタマには謎がある

田中（以下タと略）「いやー、こないだの日曜さア、原宿をぶらぶらしててさ」

丸山（以下マと略）「フンフン」

タ「竹下通りの人混みのなかを流れていたと思いなよ」

マ「ヘイヘイ」

タ「それでさ、Tシャツなんか、いっぱいぶらさがってたりしてさ」

マ「ホエホエ」

タ「おまえ、オレのことバカにしてんな！」

マ「コリヤコリヤ、と」

タ「ま、この、『うちのタマ〜』のキャラクタ商品が、けっこうあってさ、驚いたところのオレなのよ。人気あんだねー」

マ「あったりマエじゃん!! 電車に乗れば

# SMC-GAMES

わてがタマとす。  
どーぞヨロシク。  
ネコマンマほしい。



©1997 THE ANIME PRODUCTION INC.

『タマ』グッズに身をかためた女の子とかいっぱいいるしィ。ジョーシキだと思えますウ」

タ「こりゃまたシツレーいたしやした」

てな具合に大人気の『タマ』を主人公にいたしました SMC ゲーム「うちのタマ知りませんか」、お客様のご愛顧をいただきまして売行き好調で、と発売元もホクホク顔。とところがある日、その『タマ』の秘密、ダークサイド・オブ・ザ・フォースがささやかれるようになった……。

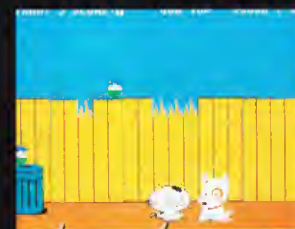
タ「ええっ! 隠しコマンドだってえ!？」

マ「シーッ。ひとに聞かれたら、どーすんのよッ!!」

タ「そっそれで、と、どんなのが…」

マ「まず、どんどん先の面へ進む」

タ「へえー」





マ「『タマ』の数を増やす…」  
 タ「うひょー」  
 マ「ボチとの再会シーンが見られる…」  
 タ「ひゃああー」  
 マ「デモゲームが作れる…」  
 タ「へー」  
 マ「『タマ』が無敵になる…」  
 タ「ぎゃああああ、うおおお——ツ」  
 マ「うるさいわねッ」  
 とゆーよーなおそるべきジジツが明らかに  
 なったのであった…。  
 タ「ね、ねえ、ねねね、ひとつだけでいい  
 からさア、教えてくれよ」  
 マ「しょーがないわねッ。じゃあ、先の面  
 へ進めるコマンドからね」  
 タ「よろしくおねげーしますだ」  
 マ「まず、ゲーム途中で[ESC]を押して…」  
 タ「ほいほい(ト、めもスル)」  
 マ「そして[カナ]+[SHIFT]+[¥]



タ「なかなかメンドーでんな」  
 マ「[カナ]を解除してから、アルファベット  
 で4文字入力したら、1から9までの数字  
 を指定すればできあがり」  
 タ「その4文字のアルファベットという  
 のは…」  
 マ「ヒ・ミ・ツ」  
 タ「そこをなんとか」  
 マ「だからア、次の面へ進むコマンドでし  
 ょう。『次』っていうのを英語にすると…。  
 キャアア——ツ」  
 タ「あ、どうしたんだ。丸山さんがさらわ  
 れちゃった、どーしたらいいんだあ、オー  
 イオイオイ」  
 いやー、こんな波乱万丈の展開になるとは  
 思わなかったぜ。隠しコマンドひとつ探し  
 出すのも命がけだねえ。まー、読者のみな  
 さんも、いろいろ試してみてください。分  
 かったら、「Oh! HIT BIT」編集部  
 に報告するのを忘れちゃいけねーぜ。



法隆寺の謎

7000円



聖徳太子一族の悲劇  
 は歴史の必然だった  
 のか。それとも、巨  
 大な闇の力が…。時  
 空を超え、壮大な謎  
 に挑むSF歴史アド  
 ベンチャーゲーム。



バイキン君のデンタルウォー

5000円



人気キャラクタ「バ  
 イキン君」ものの第  
 3弾。今回はムシバ  
 菌と対決する。虫歯  
 になった部分をブラ  
 ッシングで治してし  
 まうのがいとオカシ。



ベンジャミン

5000円



えー、おなじみベン  
 ジャミンのゲームで  
 ごさんす。いろんな  
 キャラクタの妨害を  
 かわしながら、大切  
 な鍵を拾う。グラフ  
 ィックが楽しめる。



SMC将棋

7000円



SMC用にもとうとう  
 本格的将棋ソフトが  
 登場した。美しいグ  
 ラフィックの将棋盤  
 でコンピュータと対  
 戦。強さによって対  
 局条件も選択できる。



トランジット

5000円



敵の攻撃をかわしな  
 がらユニットを運ぶ、  
 飛来する敵機をやっ  
 つける、巨大な未確  
 認飛行物体を探索す  
 る、と3ステップが  
 楽しめる複合ゲーム。



ナポレオン

5000円



トランプゲームがコ  
 ンピュータで楽しめ  
 るとうれしいね。  
 「フロッピーマガジ  
 ン」にもトランプゲ  
 ームがある。そこ  
 んとこ、頼みますぜ。



# Remain's Robot

## Making of My Robot ■ KAZ

前号でその概要を紹介したリメインズロボット。ポストロードランナーソフトとしての素質十分というわけで、今回はゲームの実際と、ロボット作りのコツについて、この道35日（何しろ発売されて日が浅いもので……）の不肖KAZがお送りしよう。

まず、（該当事者が居たらモグリだが……）「前号の記事など知らん」という人のためにざっとおさらいすると、つまりこういうことだ。

リメインズロボットはその名の通りのロボットゲームだが、「目を見張るリアルな動き」とか「3段階合体変身」とかはない。したがって、そういったソフトを手にするのが夢だという人は、①さようなら、②自分で創る、③気を取り直して読み進む、の中から選んで欲しい。ところが、この単なる駒のようなロボットと競技場からなるこのゲームは、実は将棋や碁などの高度に抽象化された戦略遊戯の持つ魅力がある。

楽しみ方は大きく分けて2つある。アクションゲームとしてのプレーヤーモード（勝手に名づけたが、カーソルパドルでロボットをあやつり応戦する）と、思考ゲームとしてのプログラミングモード（BASICライクな言語で戦闘用プログラムを作成。コンパイル後、29種ある形状パターンのひとつに搭載しオリジナルロボットとして競技場へ送り出す）である。基本アイデアはアップル用の“ROBOT WARS”なのだが、その存在を知っていてもただ指をくわえて見ているだけだった人も多いはず。ここはひとつ本家に敬意を表した上で、SMCでこの自分だけのロボットを持てるようになったことを素直に喜ぼうではないか。

はっきり言ってプレーヤーモードについては、余り書くべきことはない。これは単なるシューティングタイプのアクションゲームであり、ただ敵の動きが一筋縄ではない

のと、カーソルパドルと動きとの対応が直感的でない（意図的なものだと思われるが、前後左右が常にロボットの進行方向に対して判断されるので、退却するつもりが突撃ということがたまある）ため必勝パターンをつかむまではちょっと時間がかかりそう。しかし、プレーヤーモードのための面は16面が付属してくるに過ぎず、これらの面の突破がこのソフトの主目的ではないことを暗に示している。

さて、そこで問題のプログラミングモードである。これは、実際に自分でプログラミングをしたロボットの生死に一喜一憂した者でなければ、その醍醐味は分からない……と書いてしまっただけは話にならないが、リメインズロボットがそれだけコンピュータならではのゲームである証拠だともいえる。ロボットのプログラミングは簡単だ。だが、この際「思い通りのロボットがすぐに作れる」とは、あえて書かないことにする。これは某低価格家族電算機のユーザーがその続柄父におねだりして買ってもらったソフトとは訳が違うのだ。付属の（BASIC+C）/2に見える構造化言語を使っただけのプログラミングは、逆に通常のBASICでのプログラミングテクニックを養うためのトレーニングにもなるし、とっつき易く奥が深いという名作ゲームの条件を満たしている。

別表のサンプルプログラムは、マニュアル記載の言語仕様を読まずにすべてを理解することは難しいと思うが、数種類試作した中で比較的強かった“KAZ 2”のソースリストである。これをコンパイルし、オブジェクトをトレーサ上で動きのチェックにかけ、さらに実戦投入テストを重ねて育てて行くのだが、この“KAZ 2”も最初はあたりを見回し、敵を見ると発砲して逃げるといって、一介の臆病者に過ぎなかったわけで、まさに秘密兵器を開発するマッドサイエンティストの気分になれる。

敵を発見後とダメージを受けた時の処理は

ON～の宣言で指定される割り込みで行われ、敵に対しては先制攻撃をかけた後、被弾しないように2ブロック移動して近づき、再び元の軌道に戻って徹底攻撃をかけるというものだ。ただし近づき過ぎると敵を破壊した際の爆風でこちらも被害を受けるので、適当な距離を保つ工夫もしている（コツはリメインズロボット3原則参照）。

先程、比較的強いと書いたのは、これでも状況次第では敗れることもあるため、このソースを改造するも良し、全く異なるコンセプトで別のロボットを作るも良し、リメインズロボットを手に入れたならば、ぜひオブジェクトを持ち寄って競技会をやりたいものだ。

もちろん、戦いの場となる競技場もエディタで簡単に作れるので（こちらは「思い通りのものがすぐにできる」と書いておく）特殊な状況を先に設定して、それに合わせてプログラミングを行ったり、全状況型ロボットをオーソドックスに四角いリングで戦わせるのも意外に楽しめる。他にもプレーヤーモードの面にプログラムしたロボットを投入しての実戦テストは、ある程度のプログラムが作れるようになった中級者には良いチャレンジだし、敵ロボット無しに迷路を作り、画面に表示されるタイマを使っただけのマイクロマウス（マイクロロボットによる迷路抜け競技）のシミュレーションなど、マニュアルにない遊び方も色々できる（サンプル画面はエディタを使って描いたSonyの面とHITBITの面で単に文字として並べるのではなく、戦闘時を想定してコースを設定しているため、この面を自在に動けるロボットを設計すること自体、難しいのではないかしらん）。

いずれにしても市販のパソコンゲームでかつてこれほどクリエイティブなものはなかったといえ、ラッサピアタのBGMと合わせられることも含めて、SMC-777系ユーザー必携のソフトだろう。



## リメインズロボット3原則

### 原則 2

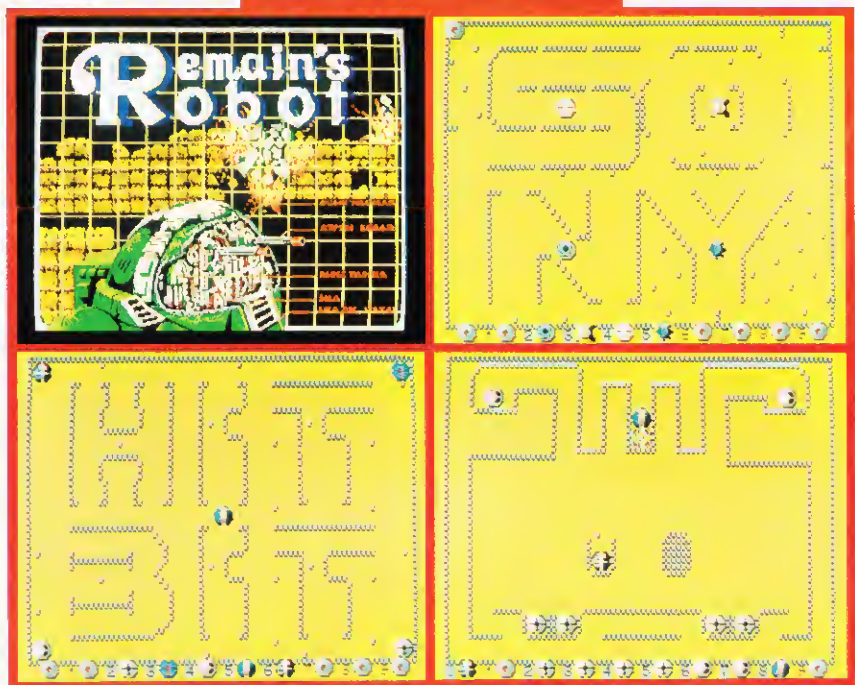
敵を見かけたらすぐに発砲すること。自分からは危害を加えない等と悠長なことを言っていてはいけません。

### 原則 1

敵を発見するため積極的に動き回ること。SLEEP命令により、敵が近づくまで、あるいは、ダメージを受けるまで活動を停止する手もあるが、双方がこのモードを同時に使用すると膠着状態となりCAUTIONを取られる（これは冗談）。

### 原則 3

上記2つの原則に違反しない限り、ロボットは自らを守らなければならない。つまり、敵破壊時や、爆弾破裂時の爆風から身を守るため、LOOK命令によって異常を察知した場合は、すみやかに退却する（時に、エネルギーがなくなりかけた敵ロボットが心中を図って接近して来る場合がある）。



```

look on
damage on
if dist==0
  look f==5 then
    goto AWAY
  endif
  if look f==1 then
    d=dist
    shoot
    if dist>d then
      go dist-d
      goto BRK
    endif
  endif
  rotate 1
  i=i-1
wend
go 1
if time>=t+5 then
  t=time
  i=i+5
  if i==4 then
    goto MV
  endif
  while i>0
    rotate -1
    i=i-1
  wend
endif
goto MAIN

LK:
d=dist
if d<30 then
  if d<6 then
    go -5
  endif
  while look f==4
    shoot
    if dist<d then
      if look r==0 then
        r=-1
      else if look l==0 then
        r=1
      else
        rotate 2
        go dist
        rotate 2
        shoot
        exit
      endif
    endif
    rotate r go 2 rotate -r
    go d+5
    rotate -r go 2 rotate -r
    goto LK
  endif
endif
wend
else
  shoot
  r=0
  if look r==0 then
    r=-1
  endif
  if look l==0 then
    r=1
  endif
  rotate r go 2 rotate -r
  go d*2/3
  rotate -r go 2 rotate r
  if look f==4 then
    goto LK
  endif
endif
endif
look on
damage on
return

DM:
rotate dd/4
goto LK

MV:
d=dist
while dist>(d/2)
  go 1
wend
goto MAIN

AS:
look off
if look b==0 then
  go -6
  if dist>5 then
    rotate 1 go 1 rotate -1 shoot
    rotate -1 go 1 rotate 1 shoot
    rotate -1 go 1 rotate 1 shoot
    goto MAIN
  endif
  if look r==0 then
    r=-1
  endif
  if look l==0 then
    r=1
  endif
  rotate r
  go dist
end 4
look on
goto MAIN

```



# F-18 誌上コンテスト 審査委員長 榎本博行



■江原隆彦氏(埼玉県朝霞市)・・・作品No.1  
A7、B9、C8、D7、E8、計39点せっかくここまでやったのだから、あとちょっと、文字等に工夫が欲しかったな

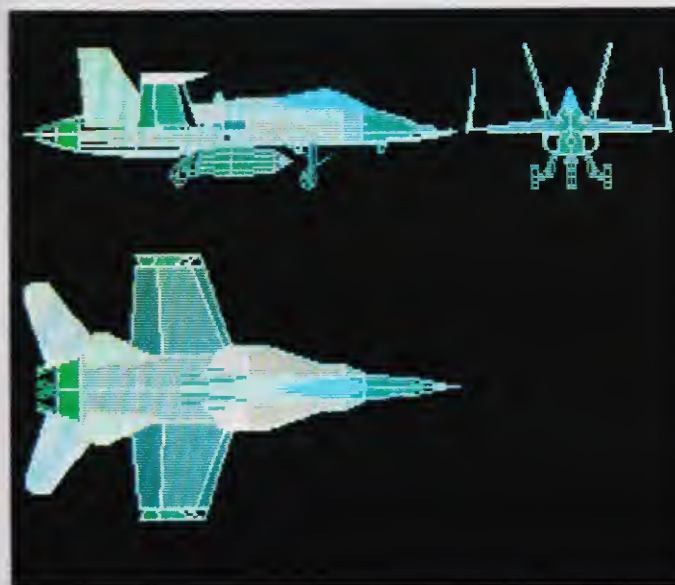


■山浦正君(東京都目黒区)・・・作品No.254  
A9、B9、C7、D7、E7、計39点もう少し、光線の具合を考慮すると、いい作品になったのに。

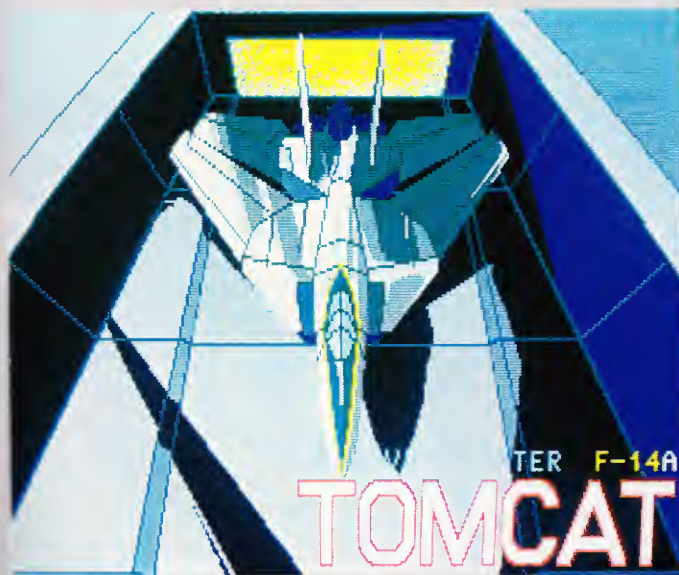




■半田慎二君(東京都杉並区)・・・作品No.5  
A6、B7、C9、D5、E7、計34点SMC  
あと、青空等の最後の詰めがあまりいようて。



■久野真樹子ちゃん(茨城県取手市)・・・作品No.6  
A6、B6、C7、D4、E7、計30点なんか制作途中という  
感じがしますが、全体的に女の子らしくていいですね



■岡田優君(東京都新宿区)・・・作品No.7  
A9、B5、C7、D6、E6、計33点F-18をF-14に仕立てたところまで  
はいいのですが、全体的なデッサンがちょっと

多数の作品の御応募ありがとうございます。厳正な審査の結果、最優秀賞は〇〇市の××君に決まりました。おめでとうございます。ウォークマン送りましたので、楽しみにしてくださいね。

…と書いてみたかった。でも、応募作品はたったの6人分。うーん、思っていたよりかなり少なかった。Oh! HITBITの発行部数からしても、少なすぎる。

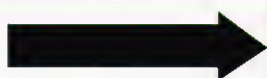
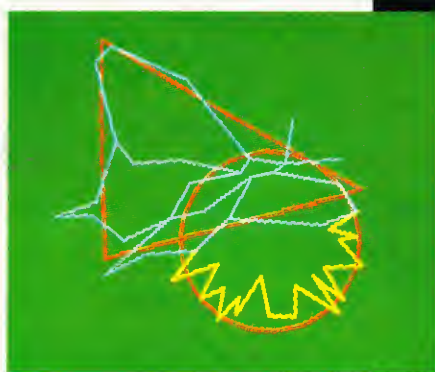
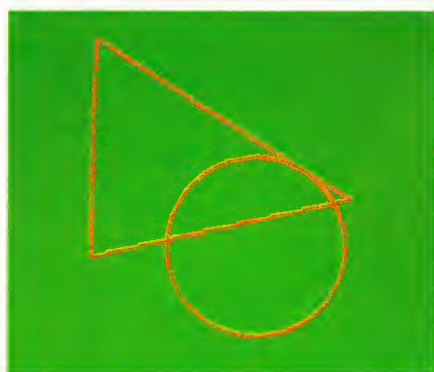
思わず、考えさせられますね。企画が悪かったのかなあ。まあ、データがかなり大きいし、いいものを描き上げるのにもかなりの日数が必要なのも分かるけど…どうしてかなあ。

とりあえず、結論から先に言ってしましましょう。今回のコンテストでは、**最優秀賞には該当作なし**。すなわち、せっかく用意しておいたウォークマンは誰も手にすることができなかったわけ。でも、もったいないので、次の企画までおあずけとします。ただ、作品を応募してくれた方々の努力は、やはり、並大抵のことではないので、応募者全員に、その作品に応じて、ゲーム・ソフトなどどっさり送ってあげたいと思います。何を送ってもらえるか、楽しみにしてください。期待はずれにはしないつもりですから。

では、応募作品の紹介にまいりましょうか。一応、作品の評価は、**A:アイデア、B:構図、C:彩色、D:背景、E:総合的な美的センス**の各10点、計50点満点の点数制で行いました。また、紹介はあくまでも応募順なので、誤解のないように。

あと、選外となった福岡市博多区の藤井茂氏。僕の作品とまったくそっくりと言うのはいただけませんね。一応、「グラフィックス・エディタを使う」の連載の主旨は、「絵のネタは提供しますが、それから先はグラフィックス・エディタを上手にを使って、自分だけのオリジナルなグラフィックスの世界を展開してください」というものなので、御了承ください。それから、Fには全く関係がないけれども、いろいろな絵を送ってくれた方々、どうもありがとう。アニメのキャラなどのように著作権にひっかかる作品以外のもので、もし、優秀な作品があれば、随時掲載したいと思っているので、ドシドシ送ってくださいね。待ってますよ。





# Exercise for Graphics Editor



陸・海・空と1シリーズ終えて、ホッとする間もなく、第4回目。思わず、「何やろうかな」などと考え込んだりするワタシです。で、思いました。Exercise for Graphics Editorなどと随分派手なタイトルがついているんですが、グラフィックス・エディタの使い方について何〜も触れていないんですね。Oh! HiTBiTの創刊号でちょっと書いていただけですから。

…という訳で、今回は、主に、グラフィックス・エディタをちゃんと使ってもらおうという観点から話を進めて行きます。

# グラフィックス・エディタを使う

## 第4回 アモルファス編

(Amorphousとは、不定形の秩序を持たない状態のことをいう。)

榎本 博行

# Exercise for Graphics Editor

## Introduction

グラフィックス・エディタ——Sony SMC-70/777でコンピュータ・グラフィックス(CG)をするのに必要不可欠なツールですが、ちゃんと使いこなしていますか。SMCが稼働するのは、ゲームをするときだけ…などという寂しいことにはなっていないでしょうね。いたんです、そ〜いうのが友だちに！ マニュアルを読んでもいまいちピンとこないというので、グラフィックス・エディタにはこりを積もらせていた罰あたりの奴が。でも、尻ごみしないで、ちょっと使ってみれば、難しくも何でもないですよ。その辺にコロがっている使えないアドベンチャーゲームよりズッと簡単。ガイドメッセージに従って、仕事を進めていだけだから。

ところで！ いきなり突然、質問。皆、グラフィックス・エディタはもっているんでしょうね。まあ、SMC-777Cにはグラフィックス・エディタが付属で付いてきちゃうんだから、大丈夫だと思いますが、もし、もっていなかったら早めにおいて方がいいですよ。そんなに高くはありませんから。

表1 グラフィックス・エディタのコマンド一覧表

モード	コマンド	意味	1	2	3
エディット	Norml	通常状態	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Block	1倍に拡大 (8×8ドット)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	2×	2倍に拡大 (4×4ドット)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	4×	4倍に拡大 (2×2ドット)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	8×	8倍に拡大 (1×1ドット)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
コマンド	Clear	画面を一色に塗りつぶす	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Color	色を塗り変える	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Load	絵をロードする	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Ovlay	絵を重ね描きする	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Save	絵をセーブする	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Line	直線を引く	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Box	四角形を描く	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	FBox	中塗りの四角形を描く	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Circ	円を描く	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	FCirc	中塗りの四角形を描く	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Fill	色を塗る	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Copy	絵をコピーする	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Text	文字・数字を書く	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Print	プリンタに出力する	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Bordr	ボーダエリアの色を変える	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Delet	PICファイルを消去する	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Renam	PICファイル名を変更する	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Edge	ふちどりをする	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Super	スーパーインポーズする	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Exit	システムに戻る	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Palet	カラー・パレットを使う	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Move	絵を移動する	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Scale	拡大・縮小する	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

- 注) 1. ソニーグラフィックス・エディター (SMW-7072: 22,800円)  
 2. SMCグラフィックス・エディター (SMW-S701D: 9,800円)  
 3. グラフィックディスク (SMC-777C: 付属品)



ところが、このグラフィックス・エディタ、同じような名前でも3種類ほどあるんですね。CP/M上のソニーグラフィックス・エディター (SMW-7072: 22,800円) と Sony Filer 上の SMCグラフィックス・エディター (SMW-S701D: 9,800円) と同じく Sony Filer 上のグラフィックディスク (SMC-777C 付属品) の3つ。あと、SMC-70用のデジタイザが使えるグラフィックス・エディターD (SMW-7072D: 35,000円) というものもありますが、ちょっと特殊なので紹介を省きます。

で、どこがどう違うのかというと、Sony Filer 版だとカラー・パレットをサポートしてあるところかな (表1参照)。多

少、機能の種類の数が違うけれども、Palet コマンド以外は他のコマンドで補えるから、ほとんど困らないといえます (が、ここでは、一応、SMW-S701Dを中心に話を進めていきます)。

でも、結局は同じグラフィックス・エディタ。使い方はまったく同じ。GREDIT **RETURN** で起動。あとは、**ESC** キーでエディット・モードとコマンド・モードをたくみに移行しながら、カーソルキー (あるいはカーソルパドル) で位置を動かし、ガイドメッセージに従ってファンクションキー **F1** ~ **F5** で機能を次々に選択していけばよろし。一口で説明してしまえばこういうことになっちゃうけど、ガイド

メッセージに表示される各コマンドの機能は省略形で表示されているので、その意味は表2を参照してください。あと、このガイドメッセージ、スペースバーを使えば表示を消したり、画面の上や下にもっていきるので便利。

ここでは、グラフィックス・エディタの機能の詳しい紹介を目的としていないので、あとは、表1と表2を参考にしながら、後に載せた Exercise を実際に実行して、自分で体験しながら使い方を覚えてくださいな。分からないときは **H** キーを押せば、グラフィックス・エディタが英語で教えてくれるから、心配御無用 (ただし、グラフィックディスク (付属品) にはこの機能があり

表2 ガイド・メッセージに表示されるコマンドの諸機能

命 令	意 味	備考
All	そのコマンドの実行範囲を全画面とする。	2
Bkgnd	(Copy, Moveコマンド実行時において,) 透明色となる色を指定する。	
Cancl	今おこなったコマンドの実行を取り消す。	1
Centr	(現在マーカのある位置に,) 円の中心を移動する。	
Color	(そのコマンドが実行する) 色を指定する。	
Cornr	そのコマンドの実行範囲を四角形の対角線をもって設定する。	
Drive	ディスク・ドライブの指定をする。	
Draw	(現在マーカのある位置を基準にして,) 直線・四角形・円を描く	
Dual	(Colorコマンド実行時において,) 2色塗りをする。	
Enter	重ね描きした絵をその位置に定着させ、Canclできないようにする。	
Home	ホーム・ポジション (画面の左上) に、絵をロードする。	
Load	現在マーカのある位置を左上にして、絵をロードする。	
Messg	ガイド・メッセージの表示位置を考える	2
Move	(現在マーカのある位置に,) 直線の基点を移動する。	
No	その命令の実行を取り消す。	
Ovlay	現在マーカのある位置を左上にして、絵を重ね描きする。	
Quit	そのコマンドの実行を終了する。	
Reset	(Paletコマンド実行時において,) カラー・パレットの色を初期限定に戻す。	
Revrs	Revrs=1で絵を左右反転させる。	
Rotat	Rotat×90°で右に絵を回転させる。	
Save	Saveコマンドを実行する。	
Scale	(Scaleコマンド実行時において,) 倍率を変える。	2
Size	(Textコマンド実行時において,) 文字の大きさを変える。	
Stop	今おこなっているコマンドの実行を途中で中止する。	
Yes	その命令の実行を遂行する。	

表3 グラフィックス・エディタのカラーキー

	B	G
黒	0	K
青	1	B
緑	2	G
空	3	T
赤	4	R
赤 紫	5	P
黄	6	Y
白	7	W
暗 緑	8	1
灰 緑	9	2
朱	10	3
山 吹	11	4
灰 青	12	5
淡 青	13	6
桃	14	7
灰	15	8

▲ 注) B. BASICのカラーコード  
G. グラフィックス・エディタのカラーキー

◀ 備考) 1はSMW-7072に固有な命令  
2はSMW-S701Dに固有な命令

図1

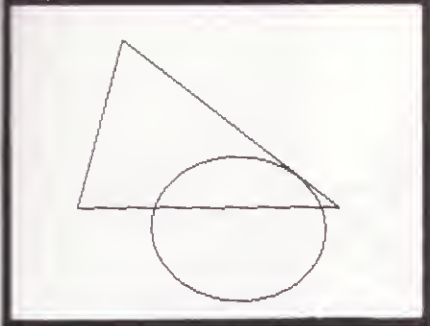
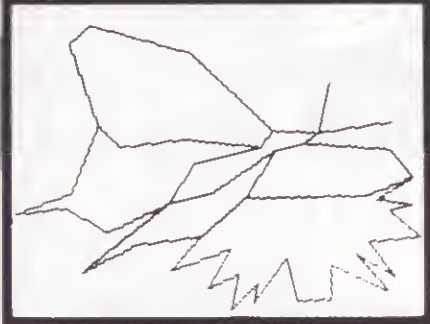


図3



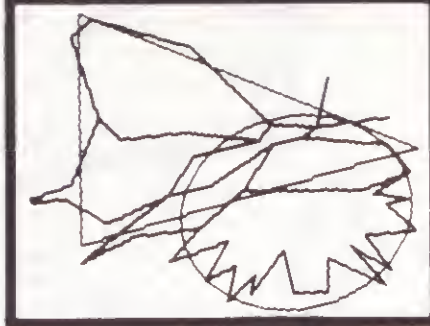
ません。あしからず)。

ただ、ちょっと注意を要するのが、色指定。色はカラーキーで入力するのだけれども、グラフィックス・エディタのカラーキーがBASICのカラーコードと違うので厄介。表3にその違いを示しておくので、参考にしてください。

## Exercise

ということなので、何よりも、まず、自分でグラフィックス・エディタを体験してみてください。演習問題として、カラー・ページにあるような蝶の絵を描いてみました。その手順を示したのがリスト1。一見するとアセンブラか何かのソース・リストのように見えますが、何のことはない、グラフィックス・エディタのコマンドを順に並べてあるだけです。要するに、グラフィックス・エディタを起動して、**[ESC]** キーでエディットモードからコマンド・モードに移行。**[↑][↓]** キーでコマンドのページをめくって、円を描くコマンド Circ **[F4]** を選択。

図2



座標 (166, 68) にカーソルパドルを使ってマーカを移動。そして、Centr **[F2]** で円の中心を設定。Color **[F3]** で **[R]** として、使う色を赤 (Red) に決める。再びカーソルパドルを使って座標 (122, 68) に移動。これが円の半径になる。Draw **[F1]** で円を描いて、Quit **[F5]** でコマンド Circを終了。…といった具合です。

あとは、この手の作業をただひたすら続ければよいのだが、一応、作業の進行具合を説明しておこう。まず、何の変哲もない画面にサクサクッと絵を描いてしまうのは常人には無理なので、大ざっぱに絵の構図を決めます。Circで花の位置、Lineで蝶の位置を指定。ここで大切なのは、使う色。ラフスケッチ用に引いた線は、あとで、Colorで消してしまうので、描こうと思う絵で使いたいような色 (ここでは赤) を使うこと。次は、Lineで描こうと思う絵をもう少し具体的に表現します。とはいっても、かなり

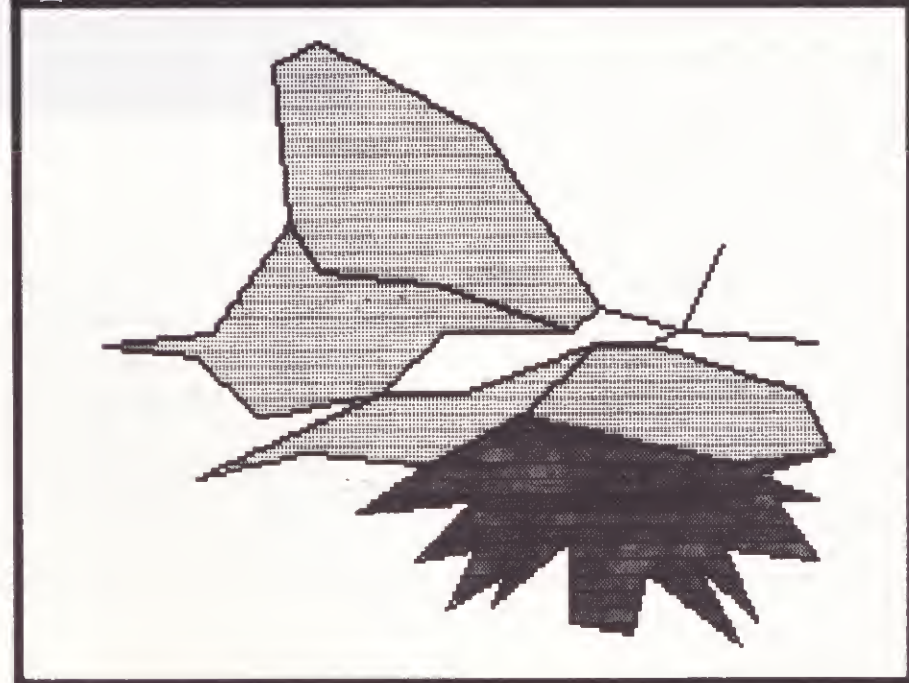
適当に。で、ここでまた重要なのは、Quitでこまめに抜け出しておくこと。これは、線を間違えて引いてしまっても、Cancelで実行を取り消せるのはいいのだが、いいところまでいった消さなくてもいい線まで一緒に消してしまうので、その予防策。あとは、Colorでラフスケッチを消して、Fillで色付け。そして、データがとんでしまわないうちに、CYOUCYO というファイル名でセーブしておく。作業状況は図1～4に示してあるので、参考してください。

と、まあ、こんな具合ですが、ここまでは誰にでもできること。**[ESC]** キーでコマンド・モードからエディット・モードに戻ってからが大変。このエディット・モードを制するかどうかで、その絵が生きたものになるか、あるいは、そのまま死んでしまうかが分かれてしまうところ。ところが、この1ドットにわたる細部の修正をここで紹介するのはちと無理。リスト1のように割とコンパクトにまとめても本1冊分になってしまう。それくらいなら、グラフィックス・データを出力して載せた方が早いですが、それではグラフィックス・エディタを使う意味がなくなってしまうことになる。そこで、ここでは、グラフィックス・エディタの秘密兵器とテクニックを挙げておくので、あとは、各自 Exercise してみてほしい。

● **[SHIFT]** キー

**[SHIFT]** キーを押したまま、同時に、カーソルパドルを使えば、マーカを早送り (8

図4





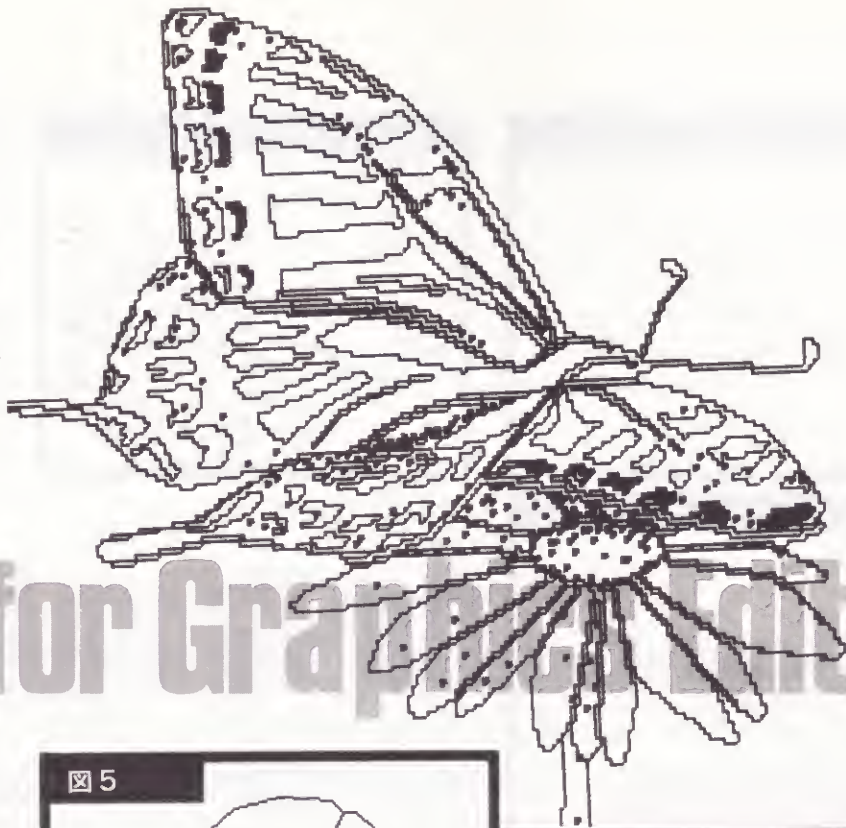
ドット単位) できる。これで、ノロノロと進むマーカの時間短縮ができる。

● **CTRL** キー

Norml 状態で、**CTRL** キーと同時にカーソルパドルを動かせば、最初マーカがあったところの色をコピーするので、これで、自由自在に曲線を描くことができる。また、ほかの状態、たとえば、Block で使えば、8×8ドットの模様をコピーできるので、いろいろなパターンを手軽に作ることができる。

● **TAB** キー

Norml 状態で **TAB** キーを押しても、座標の取り方、左上を原点にするか左下にするか変わるだけだが、ほかの状態を押せば、マス目が入るので、絵のタイトルなど描くときにキチンとしたものができ上がる。



# Application

あとは、8× 状態の使い方がいいかということでしょう。ただし、8× の画面ばかり見ていると、全体のデッサンがくろうので、こまめに Norml と 8× を往き来することです。また、8× 状態でほとんどすべての作業が足りてしまうといっても過言ではないのですが、Line, Fill, Copy コマンドは有効に使ってください。ガイドラインを引くだけでも時間が短縮できます。

最後に、すべてのテクニックを越えるものは、ガッツです。頑張ってください。

どうでしたか。うまく描けましたか。

いや～あ、実は、わたしも苦労しました、この絵には。カラーページにあるアゲハチョウは SMC-70 用に描いたので、いつものようにカラー・パレットでごまかせなくて。結局、まる3日間かかったのかな。まあ、皆さん、カラーページにある絵を真似るのもよし、ほかの蝶のパターンを描き込むのもよし、自分で満足できるグラフィックスを展開してみてください。また、「これだ!」と思う絵がありましたら、こちらの方まで送ってください。優秀な作品があれば、随

図5



図6



時、本誌に掲載させていただきます。

そんな意味もあって、今回の Application は女優&歌手編。ほとんど、編集担当者とわたしの趣味で話が進んでしまったのですが、実は、わたしは人間の顔を描くのがどうも苦手で。創刊号のときも、頑張って聖子ちゃんの顔を描いたら、そのあとすぐに、Sony からビデオ入力 of 聖子ちゃんが出てしまって、ガックリ。で、しばらく、人の顔は描かないようにしていたら、VOL.5

で落合順子さんの似顔絵を頼まれちゃって…。それから話がこじれ出して、今回に至ったわけがあります。

何はともあれ、岡田有希子ちゃんと石原真理子さんのデータを前回までと同じ要領でリスト2とリスト3に挙げておきます。ちなみにどんな顔になるかというと、図5と図6のようなナサケない顔になってしまうので、これを頑張って、本人たちに似せてあげてください。

```

ESC

Circ[F4]      (166, 68)      Centr[F2]
               Color[F3] R
               (122, 68)      Draw[F1]      Quit[F5]

Line[F1]      (130, 84)      Move[F2]
               Color[F3] R
               ( 86,180)      Draw[F1]
               ( 62, 76)      Draw[F1]
               (130, 84)      Draw[F1]      Quit[F5]

Line[F1]      (164,110)      Move[F2]
               Color[F3] T
               (135,153)      Draw[F1]
               ( 92,176)      Draw[F1]
               ( 80,170)      Draw[F1]
               ( 84,131)      Draw[F1]
               ( 92,119)      Draw[F1]
               (121,116)      Draw[F1]
               (157,104)      Draw[F1]      Quit[F5]

Line[F1]      (185,104)      Move[F2]
               (164,110)      Draw[F1]
               (157,104)      Draw[F1]
               (124,104)      Draw[F1]
               (108, 89)      Draw[F1]
               (129, 89)      Draw[F1]
               (163,101)      Draw[F1]
               (177,101)      Draw[F1]
               (185,104)      Draw[F1]      Quit[F5]

Line[F1]      (177,101)      Move[F2]
               (216, 89)      Draw[F1]
               (224, 74)      Draw[F1]
               (200, 71)      Draw[F1]
               (144, 83)      Draw[F1]
               (163,101)      Draw[F1]      Quit[F5]

Line[F1]      ( 84,131)      Move[F2]
               ( 64,104)      Draw[F1]
               ( 36,101)      Draw[F1]
               ( 60, 98)      Draw[F1]
               ( 76, 83)      Draw[F1]
               (108, 89)      Draw[F1]
               ( 60, 68)      Draw[F1]
               ( 80, 74)      Draw[F1]
               (119, 71)      Draw[F1]
               (144, 83)      Draw[F1]      Quit[F5]

Line[F1]      (196,125)      Move[F2]
               (185,104)      Draw[F1]
               (220,101)      Draw[F1]      Quit[F5]

Line[F1]      (224, 74)      Move[F2]
               Color[F3] Y
               (204, 68)      Draw[F1]
               (220, 62)      Draw[F1]
               (209, 62)      Draw[F1]
               (220, 47)      Draw[F1]
               (196, 50)      Draw[F1]
               (204, 32)      Draw[F1]
               (188, 47)      Draw[F1]
               (200, 26)      Draw[F1]
               (176, 44)      Draw[F1]
               (172, 29)      Draw[F1]
               (156, 32)      Draw[F1]
               (156, 52)      Draw[F1]
               (136, 32)      Draw[F1]
               (140, 44)      Draw[F1]
               (124, 35)      Draw[F1]
               (132, 50)      Draw[F1]
               (116, 47)      Draw[F1]
               (131, 62)      Draw[F1]
               (101, 59)      Draw[F1]
               (119, 71)      Draw[F1]      Quit[F5]

Color[F2]     All[F4] R 1      Quit[F5]

Fill[F1]      (127, 71) Y
               (127, 95) K      Quit[F5]

Color[F2]     All[F4] 1 W      Quit[F5]

Fill[F1]      ( 95, 63) 1      Quit[F5]

Save[F3]      RETURN
               CYOUCYO RETURN
               All[F4]

ESC

```

# Exercise for Graphics Editor



# Exercise Graphics Editor

リスト2 J-2

```

10 '*****
20 '*
30 '* Exercise for Graphics Editor : Mariko ISHIHARA *
40 '*
50 '* for Oh!HiTBit VOL.6 *
60 '*
70 '***** by H.ENOMOTO ***
80 '
90 '
100 WIPE
110 '
120 DIM X(400),Y(400)
130 '
140 RESTORE *MARIKO
150 FOR I=1 TO 321 : READ X(I),Y(I) : NEXT I
160 '
170 FOR I=1 TO 321 : X(I)=X(I)*5/4 : Y(I)=Y(I) : NEXT I
180 '
190 GOSUB *DISPLAY
200 '
210 END
220 '
230 *DISPLAY
240 GLOCATE( 0, 0),0,0
250 FOR K=1 TO 321
260 IF X(K)+Y(K)=0 THEN GOTO *LOOP
270 IF X(K+1)+Y(K+1)=0 THEN GOTO *LOOP
280 LINE(X(K),Y(K))-(X(K+1),Y(K+1)),7
290 *LOOP : NEXT K
300 RETURN
310 '
320 *MARIKO
330 DATA 1,131, 5,125, 17,118, 23,111, 31,89, 35,83
340 DATA 41,78, 47,67, 54,49, 57,42, 59,38, 81,20
350 DATA 183,15, 126,17, 131,22, 137,22, 145,29, 158,37
360 DATA 168,43, 165,85, 172,105, 180,129, 0,0
370 DATA 82,37, 78,40, 69,50, 69,54, 61,68, 57,80
380 DATA 56,89, 55,97, 55,101, 54,104, 57,117, 61,130
390 DATA 67,138, 77,143, 85,144, 100,140, 111,135, 116,131
400 DATA 123,123, 127,117, 130,111, 133,104, 134,96, 137,82
410 DATA 135,70, 0,0
420 DATA 131,22, 129,27, 123,35, 121,40, 121,44, 126,47
430 DATA 142,54, 152,62, 158,72, 160,82, 159,90, 156,97
440 DATA 148,112, 141,131, 138,144, 138,161, 142,176, 157,200
450 DATA 0,0
460 DATA 126,47, 131,52, 133,58, 133,64, 135,70, 140,76
470 DATA 148,92, 149,105, 148,112, 141,131, 131,152, 123,169
480 DATA 120,179, 118,200, 0,0
490 DATA 134,96, 135,102, 139,109, 141,115, 141,125, 137,132
500 DATA 134,135, 130,140, 125,149, 123,158, 123,169, 0,0
510 DATA 123,158, 111,176, 107,181, 94,200, 0,0
520 DATA 141,115, 139,117, 139,121, 138,125, 135,127, 132,126
530 DATA 130,123, 130,119, 132,116, 136,115, 139,117, 0,0
540 DATA 130,121, 127,121, 127,117, 0,0
550 DATA 135,134, 133,131, 131,130, 128,131, 124,136, 122,141
560 DATA 123,146, 124,149, 0,0
570 DATA 134,135, 133,133, 131,132, 129,135, 130,140, 0,0
580 DATA 133,124, 135,127, 134,131, 133,133, 0,0
590 DATA 132,126, 131,130, 0,0
600 DATA 123,123, 120,133, 114,142, 103,153, 99,160, 96,169
610 DATA 98,176, 102,180, 107,181, 101,177, 99,171, 100,164
620 DATA 107,154, 104,159, 102,166, 104,172, 111,176, 0,0
630 DATA 121,40, 114,42, 99,38, 82,37, 61,39, 57,42
640 DATA 54,49, 53,60, 56,70, 55,77, 50,83, 37,89
650 DATA 34,92, 32,99, 33,107, 37,115, 34,122, 32,125
660 DATA 28,127, 18,129, 11,128, 0,0
670 DATA 56,70, 57,80, 53,89, 45,100, 40,108, 34,122
680 DATA 0,0
690 DATA 55,89, 53,89, 52,95, 53,97, 55,97, 0,0
700 DATA 52,95, 51,97, 51,102, 54,104, 53,102, 51,106
710 DATA 53,111, 54,105, 54,104, 0,0
720 DATA 55,101, 52,100, 50,106, 50,116, 52,122, 55,122
730 DATA 57,117, 0,0
740 DATA 67,138, 67,153, 67,162, 67,157, 60,162, 57,168
750 DATA 59,173, 62,177, 64,188, 73,200, 0,0
760 DATA 66,65, 73,61, 76,63, 85,68, 87,73, 72,66
770 DATA 66,65, 0,0
780 DATA 103,78, 106,76, 109,75, 117,76, 121,75, 128,83
790 DATA 116,80, 108,79, 103,78, 0,0
800 DATA 70,73, 75,74, 77,75, 81,77, 83,80, 84,83
810 DATA 81,82, 78,82, 73,80, 69,75, 75,74, 0,0
820 DATA 77,75, 75,77, 75,80, 78,82, 81,80, 81,77
830 DATA 0,0
840 DATA 119,89, 115,86, 110,85, 105,86, 102,88, 105,90
850 DATA 110,92, 114,93, 118,92, 115,88, 112,86, 109,86
860 DATA 105,86, 0,0
870 DATA 109,86, 108,89, 109,91, 113,92, 115,88, 0,0
880 DATA 100,81, 96,82, 94,85, 88,102, 0,0
890 DATA 96,107, 97,109, 95,111, 0,0
900 DATA 91,110, 87,109, 85,110, 83,109, 82,107, 0,0
910 DATA 80,106, 78,106, 0,0
920 DATA 76,105, 76,102, 78,100, 0,0
930 DATA 101,118, 102,121, 102,123, 0,0
940 DATA 72,103, 66,109, 0,0
950 DATA 69,115, 72,117, 79,121, 85,122, 95,124, 0,0
960 DATA 80,118, 83,119, 0,0
970 DATA 72,120, 73,122, 81,125, 87,125, 0,0

```

```

10 /*****
20 /*
30 /* Exercise for Graphics Editor : Yukiko OKADA
40 /*
50 /*
60 /*
70 /****** by H.ENOMOTO ***
80 /
90 /
100 WIPE
110 /
120 DIM X(300),Y(300)
130 /
140 RESTORE *YUKIKO
150 FOR I=1 TO 264 : READ X(I),Y(I) : NEXT I
160 /
170 FOR I=1 TO 264 : X(I)=X(I)*5/3 : Y(I)=Y(I)*4/3 : NEXT I
180 /
190 GOSUB *DISPLAY
200 /
210 END
220 /
230 *DISPLAY
240 GLOCATE( 0, 0),0,0
250 FOR K=1 TO 264
260 IF X(K)+Y(K)=0 THEN GOTO *LOOP
270 IF X(K+1)+Y(K+1)=0 THEN GOTO *LOOP
280 LINE(X(K),Y(K))-X(K+1),Y(K+1)),7
290 *LOOP : NEXT K
300 RETURN
310 /
320 *YUKIKO
330 DATA 130, 89, 127, 92, 122,101, 118,107, 109,119, 104,123
340 DATA 100,125, 79,133, 70,133, 63,130, 58,124, 54,113
350 DATA 50,103, 50, 90, 53, 75, 56, 78, 61, 63, 67, 56
360 DATA 0, 0
370 DATA 50,103, 45, 99, 41, 87, 42, 74, 36, 61, 37, 53
380 DATA 46, 43, 71, 13, 95, 1, 118, 1, 129, 7, 133, 14
390 DATA 140, 14, 150, 19, 157, 28, 159, 34, 158, 61, 155, 71
400 DATA 149, 76, 143, 75, 138, 70, 138, 70, 135, 62, 136, 50
410 DATA 0, 0
420 DATA 155, 71, 158, 77, 160, 84, 156, 98, 148,108, 0, 0
430 DATA 165,150, 166,130, 162,118, 157,113, 148,108, 143,108
440 DATA 135,110, 120,121, 107,140, 105,150, 111,140, 121,127
450 DATA 0, 0
460 DATA 120,121, 112,121, 109,119, 0, 0
470 DATA 104,123, 100,128, 92,150, 0, 0
480 DATA 100,125, 85,144, 0, 0
490 DATA 46,150, 49,138, 54,113, 47,117, 44,121, 40,127
500 DATA 32,150, 0, 0
510 DATA 44,121, 30,124, 17,129, 9,136, 3,150, 0, 0
520 DATA 140, 32, 139, 41, 136, 50, 136, 50, 134, 54, 130, 65
530 DATA 128, 79, 130, 89, 138, 94, 146, 95, 154, 91, 0, 0
540 DATA 130, 89, 131, 96, 127,103, 0, 0
550 DATA 122,101, 124,102, 124,106, 120,108, 118,107, 0, 0
560 DATA 134, 55, 131, 49, 130, 39, 127, 33, 118, 25, 105, 22
570 DATA 0, 0
580 DATA 130, 65, 126, 66, 123, 60, 122, 58, 118, 54, 112, 51
590 DATA 99, 49, 91, 46, 87, 41, 86, 35, 89, 31, 0, 0
600 DATA 97, 40, 102, 39, 109, 42, 118, 54, 116, 60, 110, 69
610 DATA 103, 75, 0, 0
620 DATA 123, 60, 120, 68, 114, 77, 0, 0
630 DATA 112, 51, 96, 72, 0, 0
640 DATA 99, 49, 93, 54, 78, 58, 0, 0
650 DATA 85, 39, 80, 34, 75, 34, 72, 38, 0, 0
660 DATA 91, 46, 76, 55, 67, 56, 61, 53, 60, 45, 62, 38
670 DATA 0, 0
680 DATA 61, 63, 55, 63, 41, 56, 39, 58, 0, 0
690 DATA 55, 63, 51, 66, 48, 66, 42, 66, 0, 0
700 DATA 53, 75, 51, 74, 47, 73, 46, 69, 48, 66, 51, 66
710 DATA 54, 67, 54, 72, 51, 74, 0, 0
720 DATA 73, 66, 80, 66, 92, 73, 94, 78, 73, 66, 0, 0
730 DATA 105, 84, 108, 82, 119, 84, 122, 86, 123, 90, 105, 84
740 DATA 0, 0
750 DATA 72, 73, 77, 73, 82, 75, 85, 78, 86, 82, 79, 81
760 DATA 77, 80, 73, 77, 72, 73, 0, 0
770 DATA 82, 75, 82, 78, 80, 80, 77, 80, 75, 77, 77, 73
780 DATA 0, 0
790 DATA 82, 85, 80, 85, 76, 85, 0, 0
800 DATA 102, 91, 106, 88, 111, 89, 115, 91, 117, 95, 112, 96
810 DATA 106, 94, 102, 91, 0, 0
820 DATA 111, 89, 112, 91, 111, 93, 107, 93, 105, 91, 106, 88
830 DATA 0, 0
840 DATA 103, 95, 105, 97, 108, 98, 0, 0
850 DATA 100, 88, 97, 89, 90,103, 0, 0
860 DATA 96,111, 94,109, 92,111, 89,111, 83,110, 0, 0
870 DATA 88,107, 76,104, 75,101, 76, 99, 0, 0
880 DATA 75, 98, 70,100, 0, 0
890 DATA 88,119, 77,116, 67,109, 0, 0
900 DATA 81,115, 78,113, 0, 0
910 DATA 81,123, 78,123, 71,119, 70,117, 0, 0

```



# AREA-X

## 突然出現奇想天外殺伐至極爆走指向性

わっはっは！ 気分はもう戦争の高速シューティングゲームの登場である。

いやあ久しぶりですみなさん。なんと9か月ぶりですか。長いようで短いものであった。この9か月間というもの、決して遊んでいたわけではないのだよ。前回約束した STRIKER の商品化のため、バージョンアップや売り込みをしていたのだ。ところが、世の中厳しいもので、どこも商品化してくれない。物理的に不可能なテクニックを要求したり、ゲームはつまらなくてもよいからキャラクタにこれなどという会社ばかりなのだ。確かにゲームを買う人は画面写真ぐらいしか見ないで買うから、キャラクタは美しい方がいい。しかし、そのゲームがつまらなかったら、立派なサギじゃないか。それともだまされるユーザーが悪いとでもいうのだろうか。しかもよく考えてみると、そんなことは販売戦略なのであってプログラマに押しつけることではないような気がする。プンプン。

このような次第で、オレの実力が至らないばかりに STRIKER の商品化はウヤムヤになってしまった。期待しておられた読者のみなさんにはまことに申しわけない。最後の望みとしては、Oh! HiTBiT のディスク版があるが、これもどうなるかはっきりしたことはいえない。もし、どうしても STRIKER を買いたいなら、ユーザーが目ぼしい会社に商品化のお願いハガキを出すのが一番！ ってか。

とにかくあれだけ苦勞したソフトが認められなかったのだ、今機嫌が悪いのだ。だからといって読者にヤツ当たりをしてもしよ

うがないが、今回のゲームは STRIKER に比べると安直な作りになってしまった。安直さのアラワレとして、ほとんどのキャラクタが、Sony のグラフィックス・エディタで作られている。これは、キャラクタの製作を容易にすると共に、キャラが気に入らないといったクレームを聞かずにすませるためにとった処置である。つまり、キャラが気に入らないなら自分で直せどうだ難しいだろうゲーム作りの難しさ少しは思い知ったか式のメッセージなのだ。また、キャラクタデータの入力面倒な人は、ダンプリストを入力する代わりにグラフィックス・エディタで入力することもできる。ゲームの内容もいたって単純で、ギャラクシアンもどきといったものである。

ただここでカン違いして欲しくないのは、安直な作りが決して手抜きを意味するのではないということだ。前回の、エネルギーをありったけつぎ込んでゲームを作る方法を、エネルギーをできるだけ節約しつつ面白いゲームを作るといった方法に変えただけである。実際に前回より高度なテクニックも数か所で使っている。

そんな訳で、とにかくにも SMC オリジナルゲーム第3弾の登場である。

## ストーリー

時はIRGENDWANN、所はIRGENDWO。オレのスペースシップはIRGENDWIE航法で進んでいた（ネタが相当流行おくれのような気がするが、どうせ話の内容は数百年後）。ちょうどクリスタルエリアを通過しているときだった。突然他の船のSOSが飛びこんできた。こんな場所で通常空間に戻るのは自殺するようなものだが、SOSを無視した

ことが航宙管理局にバレたら免許停止処分は確実だ。オレは仕方なくIRGENDWIE機関をカットし、通常空間に戻った。

救命ポッドの収容はすぐに終わった。SOSの発信者はライセンスを取りたての若者だった。旧式の船を回航中にトラブルが起き、誤ってクリスタルエリアに出てしまったらしい。その後のことは聞くまでもない。この船だって今すぐ脱出しなければ、彼と最初にあってしまうのだ。

オレはできるだけ早く通常空間からオサラバするつもりだったが、IRGENDWIE機関が臨界に達する前にクリスタルどもがやって来た。

クリスタルエリア通称エリアXは、クリスタルどもが活動している空域のことである。クリスタルどもは、われわれの船を見つけると、ただちに仲間を集め、周囲をうろつき、体当たりを仕かけてくる。ヤツらの名前の由来は、その構造の50%以上を占めるシリコンの結晶である。そして、われわれに分かっているのはそれだけだ。クリスタルを異種生命体と信じている心理学者達は、体当たりがコミュニケーションの一形態だというし、軍人の目には高度な自己防衛システムとして映らしい。どちらも同じ位信用できないが、いずれにしても今オレ自身が危険にさらされているのは確かだ。オレは船のレーザーシステムを作動させた。

クリスタルどもは無数に現れた。時折やって来る上級クリスタルを破壊すると、攻撃は一時的に止むが、すぐに別のクリスタル群がやって来る。どうやらここはクリスタルエリアの中央付近らしい。数時間はもちこたえられたが、わずかなスキをつかれて船は大破した。緊急システムが展開した救命ポッドの中で、オレはクリスタルエリアをただようことになった。

クリスタルどもも救命ポッドには関心がな  
いようで、オレには考える時間が十分あっ  
た。せっかく救助したパイロットの行方も  
気になったが、今となってはどうしようも  
ない。やがてオレの目の前に貨物船がやっ  
てきた。オレの SOS を受信したのだろう。  
ご苦労なことだ。  
今度のパイロットは救命ポッドの扱いに不  
慣れなようで、収容にはだいぶ時間がかか  
った。船長にあいさつをするつもりだった  
が、警報にジャマされた。またクリスタル  
どもだ…。

## 入力方法

今回は、キャラクタをグラフィックス・エ  
ディタでエディットできるようにしたので、  
入力するデータはふたつある。また、グラ  
フィックス・エディタはディスクを使うため、  
CP/M か Sony FILER が必要となってい  
る。

CP/M を使う人は、DDT か Oh! HiTBiT  
Vol.3 のマシン語モニタが入力に使える。  
どちらも、入力方法を説明する必要はない  
だろう。リスト 1 は AREA-X、COM の名  
前で、リスト 2 は AREA-X、PIC のファ  
イルネームでセーブするとよい。

Sony FILER を使う場合は、DEBUG か  
Vol.3 のマシン語モニタが使える。Sony  
FILER でマシン語モニタを使う場合、リス  
ト 1、リスト 2 はそれぞれ

AREA-X. EXE, AREA-X. PIC

のファイルネームでセーブする必要がある。  
そのほかの使い方は、やはり Vol.3 を参照  
して欲しい。

DEBUG を使う人は、この続きをよく読ん  
で入力して欲しい。

### 手順 1

DEBUG コマンドが入っていてしかも 26  
K バイト以上エリアが残っている SONY  
FILER のディスクettを用意する。以後こ

のディスクettのみを用いて入力を行う。  
ディスクettをドライブに入れてリセット  
ボタンを押す。メニュー画面の黒ワクを  
DEBUG の文字に重ねて RETURN キー  
を押す。以上で DEBUGGER が起動する。

### 手順 2

次は入力である。手順 1 が終わると、  
DEBUGGER の入力ができるので、P コマ  
ンドで入力をする。リストは、連続して 16  
バイトのデータが横一列に並んでいるが、  
ここでは 1 バイトずつ入力する。画面 1 が

入力しているところである。

### 手順 3

途中まで入力したデータをセーブする例が  
画面 2 である。01 FF まで入力が終わった  
状態でセーブしている。アドレス 0200 でピ  
リオドを入力すると入力モードを抜け出す。  
あとは Y と S コマンドでセーブするだけ  
である。

### 手順 4

入力を再開するときは、入力用のディスク  
を用意した後、手順 1 と同じようにして  
DEBUGGER を起動する。その後は、画面  
3 のように Y と R コマンドで前回入力し  
たファイルをリードする。そして P コマン  
ドで入力を続ける。P コマンドでのアドレ  
ス指定が 0200 になっているのは、前回手順  
3 で入力を止めたアドレスが 0200 だから  
である。

### 手順 5

リストのデータをすべて入力できたら、ピ  
リオドを入力しコマンドモードに戻る。そ  
の後は手順 3 とまったく同じ操作をすれば  
よい。

### 手順 X

AREA-X. EXE ファイルが入力できたら、  
AREA-X. PIC ファイルも同じ要領で  
入力する。ただし、手順 3、4 の Y コマン  
ドでは

-Y AREA-X. PIC

と入力し、手順 3 の S コマンドでは

-S 0100, 42 FF

と入力する必要がある。

グラフィックス・エディタをもっている人  
は、リスト 2 を入力する代わりに、グラフ  
ィックス・エディタを使ってオリジナルキ  
ャラクタを作ることができる。このとき注  
意すべき点は次の 4 つである。

- ①キャラクタの位置を間違えないこと
- ②SAVE するときは ALL を指定するこ  
と

画面 1

```
-p 0100 C3
0100 01 C3
0101 9D 51
0102 26 01
0103 C3 41
0104 06 52
0105 01 45
0106 C5 41
0107 3A 2d
0108 07 58
0109 00 20
010A 3D 46
010B 90 4f
010C F5 52
010D 3D 20
010E 21 53
010F 00 4d
0110 02 43
0111 54 2d
0112 5D 37
0113 09 37
```

画面 2

```
01ED 00 f5
01EE 00 61
01EF 00 68
01F0 00 01
01F1 00 60
01F2 00 0f
01F3 00 09
01F4 00 30
01F5 00 04
01F6 00 01
01F7 00 30
01F8 00 80
01F9 00 9
01FA 00 4c
01FB 00 45
01FC 00 7a
01FD 00 fe
01FE 00 bd
01FF 00 38
0200 C3 #
-yar x'exx
s 0100'19H
```

画面 3

```
y are x'exx
-r
Next=1980
NextIM=1980
p 0200
0200 E0 e0
0201 CD cd
0202 8E 8e
0203 0B 0b
0204 CD cd
0205 B2 b2
0206 0B 0b
0207 01 01
0208 11 11
0209 18 18
020A 16 16
020B 08 08
020C 21 21
020D CA ca
020E 0C 0c
020F 7E 7e
0210 ED ed
0211 79 79
0212 23 23
```





③ LOAD するときは HOME を指定すること

④ ファイルネームは AREA-X. PIC とすること

また、エディットする前には、バックアップファイルを作っておいた方がよい。

#### 遊び方

さて、ここでゲームの遊び方を説明しておこう。キミは、8 方向に移動可能な小型艇をカーソルパドルで操り、クリスタルどもの体当たりやミサイル攻撃をかわしつつ、スペースバーでレーザーを発射し、クリスタルどもを撃たねばならない。20 個のクリスタルを破壊すると上級クリスタルが画面下方から現れ、こいつを撃つと次のレベルへと進める。

このように、ルールはいたって簡単である。しかし、敵の攻撃パターンは 8 種類あり、それを組み合わせることで 32 のレベルを用意してある。さらにミサイルのパターンを合わせると、144 ものレベルがあることになる。どのパターンも、油断できない手ごわいものばかりだ。

また、今回はデモ中に使用できる特殊キーをいくつか用意した。

#### RETURN キー

これは、LEVEL 1 からゲームを普通にスタートさせるキーである。

#### HOME キー

前回のゲームの続きができる。INS/DEL キーと組み合わせると、好きな LEVEL からスタートできる。

#### INS キー

HOME キーで始める LEVEL を設定する。INS キーで、LEVEL の表示が増す。

#### DEL キー

HOME キーで始める LEVEL を設定する。DEL キーで LEVEL の表示が減る。

#### CLR キー

LEVEL を 100 に設定する。

たとえば、LEVEL 50 よりスタートしたいときは、INS/DEL キーを使い LEVEL の表示を 50 にした後で HOME キーを押す。LEVEL 表示は、100 の位が表示されないので注意がいる。どの LEVEL にいるか分からなくなったときには、RETURN キーで LEVEL 1 に戻すとい。

#### プログラムの肖像

今回は、ゲームの制作をするためにオレが使っている開発環境について教えてあげよう。

ハードは、パレット付き SMC-777 に増設ディスクドライブを付け、専用ディスプレイを使っている。

ソフトは、CP/M 上で P-EDIT クラスのエディタと M 80.COM, L 80.COM だけしか使っていない。M 80, L 80 は SMC 用 FORTRAN などのディスクに入っているの、入手は簡単である。

CP/M ではさまざまな言語が使えるし、ゲーム制作以外にも開発には有効である。本格的に SMC を活用したい人は、ぜひそろえるべきだ。

なお、CP/M を使うときには、アスキー出版の実習 CP/M と応用 CP/M が参考になる。特に応用 CP/M はマシン語から CP/M のサブルーチン呼び出す方法などが詳しく書かれている。Sony FILER も CP/M ver 1.4 と同じ機能をもっているの、買っても損はしないだろう。



リスト1

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	* Ascii	Dump *
000:	C3	51	01	41	52	45	41	20	58	20	46	4F	52	20	53	4D	7A	70, AREA-X FOR SM	
001:	43	2D	37	37	37	20	56	45	52	20	31	2E	30	50	52	93		C-777 VE R 1.0 PR	
002:	47	47	52	41	4D	4D	45	44	20	26	20	43	4F	50	52	52	3F	GRAMMED & COPYR	
003:	49	47	48	54	20	42	59	20	48	49	52	4F	53	48	49	20	3D	IGHT BY HIROSHI	
004:	41	48	41	54	20	20	31	39	38	20	31	2F	33	30	30	3A		NAKATA 1 985 1/30	
005:	7E	0E	0F	11	D4	0C	0D	05	0F	FE	20	30	11	6B	01	125		...P... .. 0.0h.	
006:	0E	0F	0D	05	0C	0D	00	07	0A	49	20	63	61	6E	15	25		...?... ..I can	
007:	6E	6F	74	20	66	69	6E	64	20	22	41	52	45	41	20	58	F2	not find "AREA-X	
008:	50	49	43	22	20	66	69	6C	65	20	21	24	11	00	00	62		"PIC file 18...	
009:	21	00	40	E5	CD	FE	08	B7	28	0F	5C	FD	FE	08	B1	77	03	...	...
010:	23	10	FC	18	EF	E1	01	A0	00	09	E5	01	00	80	B7	ED	08	...	...
011:	42	E1	38	DF	01	18	0A	ED	41	0C	06	20	ED	41	01	20	0C	...	...
012:	84	ED	41	01	23	00	ED	41	3E	33	21	79	08	32	38	00	14	...	...
013:	22	39	01	01	21	01	ED	41	56	F8	11	00	40	01	80	BC	...	...	...
014:	00	2E	A0	1A	ED	79	13	04	20	01	0C	20	F5	61	68	9D		...Cy... .. 1.0sh	
015:	01	60	0F	09	30	04	01	A0	80	09	4C	45	7A	FE	BD	38	D5	...	...LEZ#B

Sum : 41 CB 1A CB C4 02 F7 42 12 BC DC F5 9E 38 3E AC 14A

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	* Ascii	Dump *	
0200:	E0	CD	BE	08	CD	B2	08	01	11	18	16	08	21	CA	0C	7E	BD	...	...	...
0210:	ED	79	23	04	15	20	F8	01	00	3C	21	00	B0	1E	16	14	D0	...	...	...
0220:	0A	7E	ED	79	23	04	04	20	1D	0C	15	20	F4	E5	61	68	1D	...	...	...
0230:	01	3C	00	09	4C	45	E1	01	0C	1D	20	F7	3E	17	1E	04	A3	...	...	...
0240:	07	1E	3C	ED	79	04	20	01	0C	1D	20	F7	3E	07	1E	10	ED	...	...	...
0250:	ED	79	04	20	01	0C	1D	20	F7	3E	07	1E	10	ED	79	04	A8	...	...	...
0260:	20	01	0C	1D	20	F7	15	20	D6	3E	03	32	F9	0C	3E	80	A2	...	...	...
0270:	32	FD	0C	3E	15	32	FC	0C	DD	21	2A	0E	2E	00	26	40	92	...	...	...
0280:	BD	75	00	DD	74	01	26	08	DD	75	02	DD	74	03	CD	6A	18	...	...	...
0290:	0F	5E	0C	32	D2	0C	DD	0C	DD	44	0B	DD	0A	08	CD	15	A7	...	...	...
0300:	15	08	CD	CF	08	CD	26	08	CD	7D	04	3E	01	CD	85	08	A9	...	...	...
0310:	3A	D2	0C	FE	14	28	0C	FE	DD	20	D6	3E	01	32	FA	0C	D6	...	...	...
0320:	32	FB	0C	CD	44	08	CD	0A	08	CD	15	08	3E	00	32	FF	20	...	...	...
0330:	0C	21	00	00	22	00	DD	0C	EE	06	21	3A	0E	11	3B	0E	1E0	...	...	...
0340:	01	2F	01	36	00	ED	B0	CD	57	08	3E	00	32	FE	0C	DD	17A	...	...	...
0350:	4B	1B	CD	6A	0F	3A	FD	0C	FE	B0	20	56	3A	F9	0C	D6	F2	...	...	...

Sum : E0 85 A9 42 D7 B5 E2 76 F9 3C 55 4E 90 9D 73 00 17C

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	* Ascii	Dump
0500:	01	27	32	F9	0C	CD	35	08	DD	21	2A	0E	21	0C	40	DD	E2	...	...
0510:	75	08	DD	74	09	2E	00	26	40	DD	75	00	DD	74	01	26	35	...	...
0520:	08	DD	75	02	DD	74	03	06	20	C5	DD	21	2A	0E	DD	6E	EC	...	...
0530:	02	DD	66	03	25	DD	75	02	DD	74	03	06	20	4A	08	DD	A0	...	...
0540:	08	CD	15	08	CD	CF	08	CD	26	08	3E	01	CD	85	08	C1	F1	...	...
0550:	10	D7	3E	15	32	FC	0C	3E	FE	32	FD	0C	3E	DD	26	08	DD	...	...
0560:	C3	04	CD	97	05	CD	C7	0F	CD	44	10	CD	6C	07	CD	1D	1A	...	...
0570:	06	CD	44	08	CD	A0	08	CD	15	08	CD	CF	08	CD	26	08	DD	...	...
0580:	00	00	18	3E	01	CD	85	08	3A	FD	0C	FE	00	2E	33	00	15	...	...
0590:	D2	0C	FE	0C	3A	59	4A	3A	FC	0C	B7	20	C2	3E	00	3A	57	...	...
0600:	21	3A	0E	11	10	00	06	13	DD	B6	03	DD	19	10	F9	15	...	...	
0610:	20	AD	21	FA	0C	34	3A	FC	0C	B6	03	DD	19	10	F9	15	...	...	
0620:	0A	02	CD	C3	04	CD	97	05	CD	C7	0F	CD	44	10	CD	6C	07	...	...
0630:	06	CD	44	08	CD	A0	08	CD	15	08	CD	CF	08	CD	26	08	DD	...	...
0640:	00	00	18	3E	01	CD	85	08	3A	FD	0C	FE	00	2E	33	00	15	...	...
0650:	DD	21	3A	0E	11	10	00	06	13	DD	B6	03	DD	19	10	F9	15	...	...
06F0:	DD	21	3A	0E	11	10	00	06	13	DD	B6	03	DD	19	10	F9	15	...	...

Sum : 9B 41 F6 91 B2 28 79 56 6F E9 FC 1D D0 50 6D 79 :B3

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	* Ascii	Dump *
0A00:	87	20	BF	3A	FD	0C	FE	00	28	B8	3A	F9	0C	B7	C2	EF	:5E	キ ヲ人ノ頭、 (7)キ、サ	
0A10:	02	21	1E	0D	11	3A	0E	01	80	00	ED	B0	3E	08	CD	BE	:F9	、.....、	

Sum : EB 35 9F 5B 4D BB 21 E5 A6 66 51 A4 66 CB 19 B4 :F4

0500:	02	B7	ED	52	DD	75	02	DD	74	03	18	15	DD	6E	02	DD	:F7	、xor、sun、t、...	
0510:	66	03	7C	FE	C8	30	0A	11	00	02	19	DD	75	02	DD	74	:B6	、t、xor、o、...	
0520:	03	01	51	01	ED	78	C9	47	28	18	CD	4F	28	30	1D	E6	:8F	、t、xor、o、...	
0530:	E6	04	C8	B8	1B	E6	04	C8	B8	A2	0C	FE	16	28	1B	FE	:C4	、xor、o、...	
0540:	19	C0	DD	6E	00	DD	66	01	7C	FE	78	DD	11	00	01	19	:55	、xor、o、...	
0550:	DD	75	00	DD	74	C1	C9	DD	6E	00	DD	66	01	7C	FE	09	:7F	、xor、o、...	
0560:	DD	B1	00	01	B7	ED	52	DD	75	00	DD	74	C1	C9	DD	34	:5E	、xor、o、...	
0570:	0A	DD	7E	0A	FE	11	28	15	21	68	40	C8	3F	28	07	47	:04	、xor、o、...	
0580:	11	00	0A	19	10	FD	DD	75	08	DD	74	09	3E	3E	80	32	:AE	、xor、o、...	
0590:	FD	0C	DD	36	03	00	C9	3A	FB	0C	B7	28	04	30	32	F8	:70	、xor、o、...	
05A0:	0C	DD	21	3A	0E	06	03	C5	DD	7E	03	FE	10	38	19	67	:45	、xor、o、...	
05B0:	DD	6E	02	11	00	01	B7	ED	52	DD	75	02	DD	74	C1	C9	:CD	、xor、o、...	
05C0:	11	10	00	DD	19	10	E0	C9	DD	36	03	00	3A	FB	0C	B7	:DB	、xor、o、...	
05D0:	20	ED	3A	FD	0C	FE	FF	20	E6	01	51	01	ED	78	C9	B7	:3D	、xor、o、...	
05E0:	28	13	DD	18	C8	B7	28	DB	18	C8	B7	28	DB	18	3A	DA	:70	、xor、o、...	
05F0:	0C	FE	20	CD	3A	5E	02	32	FB	0C	FD	21	2A	0E	FD	66	:43	、xor、o、...	
Sum	1	85	47	1C	32	B2	95	ED	38	1B	F7	39	66	15	A4	91	B0	:31	



Addr:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
0D00:	00	00	00	00	00	01	02	03	04	05	06	07	08	10	02	90	C6	.....	.....	.....	.....
0D10:	22	99	99	99	99	0E	40	EE	44	44	44	44	40	40	10	E2	.....	.....	0DD.E	.....	
0D20:	00	01	00	00	00	02	5C	4A	00	10	00	00	00	00	00	30	9F	.....	.....	.....	.....
0D30:	00	01	40	00	00	02	5C	4A	00	10	00	00	00	00	00	50	A9	.....	.....	.....	.....
0D40:	00	01	00	00	00	02	5C	4A	00	10	00	00	00	00	00	70	F2	.....	.....	.....	.....
0D50:	00	01	40	00	00	02	5C	4A	00	10	00	00	00	00	00	10	C1	.....	.....	.....	.....
0D60:	00	07	00	00	00	0E	5C	4A	00	10	00	00	00	00	00	30	99	.....	.....	.....	.....
0D70:	00	07	40	00	00	0E	5C	4A	00	10	00	00	00	00	00	50	A3	.....	.....	.....	.....
0D80:	00	07	00	00	00	0E	5C	4A	00	10	00	00	00	00	00	70	F2	.....	.....	.....	.....
0D90:	00	07	40	00	00	0E	5C	4A	00	10	00	00	00	00	00	25	40	17B	.....	.....	.....
0DA0:	77	00	00	00	00	2D	77	77	00	00	00	0E	AD	77	00	C4	.....	.....	.....	.....	
0DB0:	00	00	00	43	58	77	00	00	00	00	7E	C4	77	00	00	00	C8	.....	.....	.....	.....
0DC0:	00	00	5F	11	77	00	00	00	7C	10	77	00	00	00	00	7A	64	.....	.....	.....	.....
0DD0:	18	77	00	00	00	00	73	94	77	00	00	00	00	00	4A	B3	77	81	.....	.....	.....
0DE0:	00	00	00	00	00	74	BB	77	00	00	00	70	75	77	00	00	02	.....	.....	.....	.....
0DF0:	00	00	4E	69	77	00	00	00	00	70	16	77	00	00	00	00	2B	.....	.....	.....	.....

Sum : B1 CF BF 8B 4C F0 51 66 E5 49 55 F6 06 BE 51 CE 17F

Addr:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
0E00:	50	AB	77	00	00	00	00	2A	AC	77	00	00	00	00	18	3D	14	P.....*	P.....*	.....	.....
0E10:	77	00	00	00	00	21	BF	77	00	00	00	00	6E	B1	77	00	134	M.....*	M.....*	.....	.....
0E20:	00	00	00	33	16	77	00	00	00	00	00	44	00	00	00	00	:04	.....*	.....*	.....	.....
0E30:	00	00	0C	40	00	00	00	00	00	00	00	00	00	00	00	00	:4C	.....*	.....*	.....	.....
0E40:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....*	.....*	.....	.....
0E50:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....*	.....*	.....	.....
0E60:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....*	.....*	.....	.....
0E70:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....*	.....*	.....	.....
0E80:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....*	.....*	.....	.....
0E90:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....*	.....*	.....	.....
0EA0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....*	.....*	.....	.....
0EB0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....*	.....*	.....	.....
0EC0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....*	.....*	.....	.....
0ED0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....*	.....*	.....	.....
0EE0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....*	.....*	.....	.....
0EF0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....*	.....*	.....	.....

Sum : C7 AB B3 73 16 9B BF A1 AC 77 00 44 6E B1 BF 3D : 9B

Addr:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
0F00:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....	.....	.....
0F10:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....	.....	.....
0F20:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....	.....	.....
0F30:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....	.....	.....
0F40:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....	.....	.....
0F50:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....	.....	.....
0F60:	00	00	00	00	00	00	00	00	00	00	00	21	06	1A	3A	FA	0C	37	.....	.....	.....
0F70:	3D	E6	1F	2B	07	47	11	0A	00	19	10	FD	DD	21	3A	0E	3C	.....	.....	.....	.....
0F80:	06	08	11	10	00	7E	DD	77	0B	23	DD	19	10	F7	7E	32	0D	.....	.....	.....	.....
0F90:	BE	16	27	7E	32	BF	16	21	00	00	22	BA	16	22	BC	16	83	.....	.....	.....	.....
0FA0:	3A	FA	0C	FE	21	DB	3C	BF	CB	3F	CB	3F	CB	3F	CB	3F	EA	.....	.....	.....	.....
0FB0:	07	20	02	3E	08	32	BF	16	21	10	00	22	BA	16	29	22	EA	.....	.....	.....	.....
0FC0:	BC	16	C9	DD	21	3A	0E	3C	BE	16	47	DD	7E	0A	E6	07	8B	.....	.....	.....	.....
0FD0:	20	04	DD	CB	08	B6	C5	DD	CB	08	7E	20	2E	DD	BB	15	6E	.....	.....	.....	.....
0FE0:	DD	CB	08	66	20	1C	FD	21	3C	16	7E	DD	BE	E6	07	2B	48	.....	.....	.....	.....
0FF0:	0B	47	11	02	00	FD	19	10	FC	DD	6E	00	FD	66	01	CD	20	.....	.....	.....	.....

Sum : 03 4A 23 02 AE 97 E9 CB 2C 4B 7F FB C6 7B B7 7B : 99

Addr:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*	
1000:	43	10	01	10	00	DD	09	C1	10	C1	C9	DD	34	0A	DD	7E	11B	.....	.....	.....	.....	
1010:	0A	FE	08	30	19	21	70	40	DD	7E	0A	E6	07	2B	07	11	3BC	.....	.....	.....	.....	
1020:	00	0A	19	3D	20	FC	DD	75	08	DD	74	09	18	04	21	00	13C	.....	.....	.....	.....	
1030:	00	DD	75	02	DD	74	03	DD	75	06	DD	74	07	DD	CB	08	10B	.....	.....	.....	.....	
1040:	BE	18	BF	E9	3A	BF	16	B7	CB	DD	21	BA	0E	47	C5	DD	1AB	.....	.....	.....	.....	
1050:	5E	10	CD	FD	01	10	00	DD	09	C1	10	F1	C9	3A	2B	13A	.....	.....	.....	.....		
1060:	0E	57	3A	2D	0E	5F	DD	7E	02	B7	28	48	DA	06	BB	28	17D	.....	.....	.....	.....	
1070:	1E	DD	6E	0D	66	07	ED	4B	BC	16	38	08	BD	87	ED	42	1E0	.....	.....	.....	.....	
1080:	DD	75	06	DD	74	07	1B	07	09	DD	75	06	DD	74	07	DD	165	.....	.....	.....	.....	
1090:	7E	01	DD	06	BA	CB	DD	6E	04	DD	66	05	ED	4B	BA	16	179	.....	.....	.....	.....	
10A0:	38	0A	B7	ED	42	DD	75	04	DD	74	05	C9	DD	75	04	1FC	.....	.....	.....	.....		
10B0:	DD	74	05	C9	3A	FC	DD	0E	07	CB	DD	21	3A	0E	06	DD	151	.....	.....	.....	.....	
10C0:	7E	08	E6	50	20	07	FD	7E	0A	E6	07	2B	0A	C5	01	10	160	.....	.....	.....	.....	
10D0:	00	FD	09	C1	10	E9	3A	FC	DD	0E	07	CB	DD	21	3A	0E	121	.....	.....	.....	.....	
10E0:	00	01	00	03	09	DD	75	02	DD	74	03	DD	75	06	DD	74	07	162	.....	.....	.....	.....
10F0:	02	01	00	06	09	DD	75	02	DD	74	03	DD	75	06	DD	74	07	156	.....	.....	.....	.....

Sum : 85 4F 52 48 3C 45 B9 22 9E 7F 46 DB F1 1B AD 5A : 1EB

Addr:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
1100:	01	92	38	09	FE	08	38	0C	21	00	FF	18	07	FE	F8	30	183	.....	.....	.....	.....
1110:	03	21	00	01	0E	DD	75	04	DD	74	05	21	00	0A	F8	7E	03	70	.....	.....	.....
1120:	93	38	09	FE	08	38	0C	21	00	FF	18	07	FE	F8	30	03	185	.....	.....	.....	.....
1130:	21	00	02	DD	75	06	DD	74	07	DD	7E	05	DD	7B	07	C0	18D	.....	.....	.....	.....
1140:	21	00	00	DD	75	02	DD	74	03	DD	75	06	DD	74	07	C3	13C	.....	.....	.....	.....
1150:	B4	10	DD	7E	03	B7	2B	3C	DD	35	0A	20	04	DD	36	0A	19A	.....	.....	.....	.....
1160:	08	21	14	0D	75	07	0A	E6	07	2B	07	11	0A	19	3D	16F	.....	.....	.....	.....	
1170:	20	FC	DD	75	08	DD	74	09	11	1E	00	3A	2B	0E	DD	96	1E5	.....	.....	.....	.....
1180:	01	30	03	11	02	FF	DD	6E	04	DD	66	05	1E	DD	75	04	1E2	.....	.....	.....	.....
1190:	DD	74	05	C9	3A	FE	B7	CB	DD	7E	21	3A	1E	DD	75	04	1E2	.....	.....	.....	.....
11A0:	7E	03	B7	2B	3C	FE	18	08	11	DD	21	00	01	ED	F0	DD	655	.....	.....	.....	.....
11B0:	E5	D1	21	4E	16	01	0C	00	00	DD	30	C6	08	DD	77	01	DD	.....	.....	.....	.....
11C0:	3F	FE	20	38	05	21	00	FF	C6	30	C6	08	DD	77	01	DD	80A	.....	.....	.....	.....
11D0:	75	04	DD	75	08	DD	74	09	11	1E	00	3A	2B	0E	DD	96	1E5	.....	.....	.....	.....
11E0:	04	DD	36	0A	19	3D	16	F	DD	75	06	DD	74	07	C3	13C	.....	.....	.....	.....	



Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
000:	D0	C0	14	00	00	00	00	00	B0	A0	18	00	B0	A0	1C	00	:78		ミク.....	-	..
010:	B0	A0	20	00	B0	A0	24	00	B0	A0	28	00	B0	A0	2C	00	:D8		..	..	..
020:	B0	A0	30	00	DF	CF	3F	00	DF	CF	3F	00	DD	CF	3F	00	:44		..	0..マ?	..マ?..マ?
030:	DC	CF	3F	00	DB	CF	3E	00	DA	CF	3E	00	D9	CF	3E	00	:9F		ウマ?..ロマ?..	レマ?..ルマ?	
040:	DB	CF	3E	00	D7	CF	3D	00	D6	CF	3D	00	D5	CF	3D	00	:8B		リマ?..ラマ?..	ヨマ?..ムマ?	
050:	D4	CF	3D	00	D3	CF	3C	00	D2	CF	3C	00	D1	CF	3C	00	:77		ハマ?..タマ?..	メマ?..ムマ?	
060:	D0	CF	3C	00	DF	C0	14	00	D0	CF	3C	00	1A	00	D0	19	:E1		ミマ?..タマ?..	ミク..ミク..	
070:	D0	C0	18	00	D0	C0	17	00	D0	C0	16	00	D0	C0	15	00	:9A		ミク..ミク..	ミク..ミク..	
080:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	.....	
090:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	.....	
100:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	.....	
110:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	.....	
120:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	.....	
130:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	.....	
140:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	.....	
150:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	.....	
160:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	.....	
170:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	.....	
180:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	.....	
190:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	.....	
Sum :	5B	FC	72	00	C3	5C	45	00	60	FC	66	00	5C	FC	6C	00	:B0				

ARMAX

## リスト2

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
0100:	0E	00	01	70	02	00	01	07	05	00	01	0E	01	E0	06	00	:84		..p.....	.....	
0110:	02	33	06	00	02	BB	06	00	01	33	01	DD	05	00	01	07	:1D		..3..7..	..3..7..	
0120:	02	77	01	70	05	00	02	22	06	00	01	BB	01	AA	06	00	:86		..w.p.....	.....	
0130:	02	77	06	00	01	06	01	60	04	00	06	77	05	00	01	44	:82		..w.....	.....	
0140:	01	64	01	44	01	64	01	44	01	64	01	44	01	64	02	00	:65		..d..d..d..	..d..d..d..	
0150:	04	77	2A	00	00	00	00	01	07	01	70	02	00	01	07	01	:36		..p.....	.....	
0160:	70	04	00	02	EE	05	00	04	33	04	00	04	BB	04	00	01	:68		..p.....	.....	
0170:	03	01	33	01	DD	01	D0	03	00	01	07	01	70	01	03	01	:67		..3..3..5..	.....	
0180:	30	01	07	01	70	03	00	04	22	04	00	01	08	01	BB	01	:9F		..0.....	.....	
0190:	AA	01	A0	04	00	01	77	02	00	01	77	04	00	01	06	01	:F8		..z.....	.....	
01A0:	60	01	06	01	00	03	00	06	77	05	00	01	44	01	64	01	:FD		.....	.....	
01B0:	44	01	64	01	44	01	64	01	44	01	64	01	00	06	77	29	:A4		..D..D..d..	..D..D..d..	
01C0:	00	00	00	00	01	07	01	70	02	00	01	07	01	70	03	00	:04		.....	.....	
01D0:	01	0E	02	EE	01	E0	03	00	01	03	01	33	02	00	01	33	:51		..0..3..5..	.....	
01E0:	01	30	02	00	01	0B	04	BB	01	B0	03	00	02	33	02	DD	:C6		..0.....	.....	
01F0:	03	00	01	70	01	00	02	33	01	00	01	07	02	00	01	02	:8B		..p.....	.....	
Sum :	0F	43	BF	8C	EE	32	C0	40	2D	5B	62	B0	BE	A0	ED	8C	:9E				

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
0200:	01	22	02	00	01	22	01	20	03	00	02	BB	02	AA	03	00	:DB		.."......	.....	
0210:	01	07	01	00	01	07	01	70	01	00	01	70	03	00	01	60	:5B		.....p.....	.....	
0220:	02	00	01	06	02	00	02	77	04	00	02	77	04	00	01	66	:6C		.....p.....	.....	
0230:	01	06	01	66	01	06	01	66	01	06	01	66	01	06	02	07	:5A		..f..f..f..	.....	
0240:	01	77	01	07	01	77	01	07	01	77	29	00	00	00	00	01	:AF		.....w.....	.....	
0250:	07	01	70	02	00	01	07	01	70	03	00	04	EE	03	00	01	:EC		..p.....	.....	
0260:	33	01	30	02	00	01	03	01	33	02	00	01	4B	04	BB	01	:AC		..3..0.....	.....	
0270:	44	02	00	01	03	02	33	02	DD	01	D0	01	00	01	07	01	:39		..D.....	.....	
0280:	00	01	03	01	33	01	D3	01	30	01	00	01	70	01	00	01	:B1		.....3..t..	.....	
0290:	22	01	20	02	00	01	02	01	22	02	00	01	08	02	BB	02	:3B		.....	.....	
02A0:	AA	01	A0	02	00	01	70	01	07	02	77	01	44	01	64	01	:44		..z.....	.....	
02B0:	00	01	60	02	00	01	06	02	00	02	77	04	00	02	77	04	:66		.....	.....	
02C0:	00	01	44	01	64	01	44	01	64	01	44	01	64	01	44	01	:44		..D..D..D..	..D..D..D..	
02D0:	64	08	77	28	00	00	00	00	01	07	01	70	02	00	01	07	:9B		..d..w.....	.....	
02E0:	01	70	02	00	01	0E	01	00	01	04	01	40	01	00	01	E0	:A8		..p.....	.....	
02F0:	01	00	01	03	01	33	04	00	01	33	01	30	01	04	01	44	:EC		.....3..	..3..0..D..	
Sum :	B6	27	B7	AB	A2	F0	E4	7E	4A	C9	34	F6	96	D0	49	07	:F6				

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
0300:	03	BB	01	B4	01	44	01	40	01	00	03	33	03	DD	01	00	:11		..7..I..D..@	.....	
0310:	01	07	01	00	01	33	03	D3	01	00	01	70	01	02	22	:AC		.....3..t..	.....		
0320:	03	00	01	22	01	20	01	00	03	BB	03	AA	01	00	01	07	:BC		.....	.....	
0330:	01	00	04	77	01	00	01	70	01	00	01	66	01	06	02	00	:5F		.....w..p.....	.....	
0340:	01	60	01	66	01	00	01	77	06	00	01	77	04	00	01	44	:0B		.....	.....	
0350:	01	64	01	44	01	64	01	44	01	64	01	44	01	64	08	77	:E2		..d..d..d..	..d..d..d..	
0360:	2B	00	00	DD	00	01	07	01	70	02	00	01	07	01	70	02	:2B		.....	.....	
0370:	00	01	EE	01	00	02	44	01	00	01	EE	01	00	01	03	01	:2C		.....	.....	
0380:	30	04	00	01	03	01	30	01	04	01	44	01	4B	02	BB	02	:BE		..0.....	.....	
0390:	44	01	40	01	03	03	33	03	DD	01	D0	01	70	01	03	01	:E6		..D..@...3..	.....	
03A0:	3D	01	D3	01	DD	01	3D	01	30	01	07	01	02	01	20	01	:8B		..t..t..t..	.....	
03B0:	22	01	20	02	00	01	02	01	20	01	0B	03	BB	03	AA	01	:E1		.....	.....	
03C0:	A0	02	07	04	77	02	70	01	06	01	00	02	60	02	06	01	:09		.....	.....	
03D0:	00	01	60	01	77	06	00	01	77	04	00	01	44	01	64	01	:06		.....	.....	
03E0:	44	01	64	01	44	01	64	01	44	01	64	08	77	2B	00	00	:A4		..D..d..D..	..D..d..w..	
03F0:	0D	00	01	07	01	70	02	00	01	07	01	70	01	00	01	0E	:11		.....p.....	.....	
Sum :	F6	92	F6	F7	1C	7D	CB	49	70	33	83	F1	A6	7D	75	FC	:ED				

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
0400:	01	EE	01	04	02	44	01	40	01	EE	01	E0	01	33	06	00	:85		.....D..@..	.....	
0410:	01	33	03	44	01	BB	01	B4	03	44	04	33	04	DD	01	70	:8C		..3..D..7..I..	..D..3..D..p..	
0420:	01	33	01	DD	01	DD	01	DD	02	03	44	01	07	01	22	00	:8D		..3..D..7..I..	..D..3..D..p..	
0430:	01	02	01	22	03	00	01	DD	04	BB	04	00	07	01	70	01	:32		.....7..I..p..	.....	
0440:	04	77	01	70	01	07	01	06	01	00	01	06	01	60	01	06	:68		.....W..p..	.....	
0450:	01	60	01	00	01	60	01	77	08	00	01	66	01	06	01	66	:1B		.....W..p..	.....	
0460:	01	06	01	66	01	06	01	66	01	06	01	77	01	07	01	77	:DB		.....f..f..	.....W..p..	
0470:	01	07	01	77	01	07	01	77	01	07	2B	00	00	00	00	01	:3E		.....W..p..	.....	
0480:	07	01	01	07	01	70	01	07	01	70	01	00	02	EE	04	5F			.....p..p..	.....	
0490:	44	02	EE	01	33	06	00	01	33	03	44	01	4B	04	44	04	:81		.....D..3..	.....3..D..B..D..	
04A0:	33	04	DD	01	73	01	77	01	7D	01	07	01	DD	01	77	01	:AD		.....3..S..W..	.....3..S..W..	
04B0:	7D	01	37	01	22	02	00	01	22	01	20	02	00	01	22	04	:47		.....2..7..	....."....."	
04C0:	BB	04	AA	01	70	06	77	01	07	01	60	06	01	06	01	06	:CE		.....7..I..p..	.....	
04D0:	77	08	00	01	44	01	64	01	44	01	64	01	44	01	64	01	:B1		.....W..p..	.....D..D..D..D..	
04E0:	44	01	64	08	07	78	00	00	00	00	02	77	01	07	01	70	:4F		.....D..d..	.....w..w..p..	
04F0:	02	77	01	00	02	EE	04	44	02	EE	01	33	06	00	01	33	:10		.....W..p..	.....D..3..	



Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F iSum \* Ascii Dump \*
0700: 0C 00 01 07 01 77 01 00 01 17 01 71 01 00 01 77 :90
0710: 01 70 02 00 04 EE 03 00 01 33 01 30 02 00 01 03 :B3
0720: 01 33 02 00 01 44 04 BB 01 01 33 01 02 00 01 02 :DE
0730: 02 0C 01 00 03 00 01 D3 01 DD 02 3D 03 00 01 22 :A9
0740: 01 20 02 00 01 02 01 77 01 02 02 00 01 02 00 :48
0750: 01 40 02 00 01 02 01 07 02 77 01 70 01 07 03 :00
0760: 01 60 02 00 01 06 02 00 02 77 04 00 02 77 04 :66
0770: 01 44 01 64 01 44 01 64 01 44 01 64 01 44 01 :6A
0780: 08 77 28 00 00 0C 00 01 07 01 77 01 02 77 01 :6A
0790: 00 77 77 01 70 02 00 01 0E 02 EE 01 E0 03 00 :01
07A0: 03 01 33 02 00 01 33 01 30 02 00 01 0B 04 BB 01 :6C
07B0: B0 03 00 02 DD 02 CC 0B 00 01 02 01 22 02 00 01 :94
07C0: 22 01 20 03 00 02 AA 02 44 03 00 01 07 01 00 01 :A5
07D0: 07 01 70 01 00 01 70 03 00 01 60 02 00 01 06 :02
07E0: 00 02 77 04 00 02 77 04 00 01 44 01 64 01 44 01 :EA
07F0: 64 01 44 01 64 01 44 01 64 01 07 06 77 01 70 :28

Sum : 5C F4 2A 39 8E 7C E2 33 FB 19 1F CA FC 8B FB 51 :CC

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F iSum \* Ascii Dump \*
0800: 00 00 0C 00 02 77 04 00 02 77 03 00 02 EE 05 00 :FA
0810: 04 33 04 00 04 BB 04 00 01 00 01 DD 01 CC 01 C0 :78
0820: 0C 00 04 22 04 00 01 0A 01 AA 01 44 01 40 04 00 :76
0830: 01 77 02 00 01 77 04 00 01 06 01 60 01 06 01 60 :C6
0840: 03 00 06 77 05 00 01 66 01 06 01 66 01 06 01 66 :CB
0850: 01 06 01 66 01 06 01 00 01 07 01 77 01 07 01 77 :76
0860: 01 07 01 77 29 00 00 0C 00 01 74 01 47 04 00 01 :77
0870: 74 01 47 03 00 01 0E 01 E0 06 00 02 33 06 00 02 :F2
0880: BB 06 00 01 DD 01 CC 0E 02 22 06 00 01 AA 01 10 :50
0890: 44 06 00 02 77 06 00 01 06 01 60 04 00 06 77 05 :B7
08A0: 00 01 44 01 64 01 44 01 64 01 44 01 64 01 44 01 :44
08B0: 64 02 00 04 77 2A 00 00 18 00 02 EE 04 00 01 03 :1B
08C0: 02 33 01 30 05 00 02 BB 07 00 01 33 01 3D 04 00 :A5
08D0: 01 07 02 77 01 70 05 00 02 22 07 00 01 BB 01 BA :99
08E0: 05 00 02 77 05 00 01 06 02 64 01 60 03 00 06 :77
08F0: 05 00 01 46 01 64 01 46 01 64 01 46 01 64 01 46 :50

Sum : FA 01 AF E5 75 B6 36 94 75 38 4E 33 EF 7B 7F B1 :1C

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F iSum \* Ascii Dump \*
0900: 01 64 02 00 04 77 2A 00 00 17 00 03 EE 04 00 01 :19
0910: 33 01 30 01 03 01 33 03 01 08 04 BB 05 00 02 71 :B3
0920: 33 01 3D 03 00 01 07 01 70 01 03 01 30 01 07 01 :2B
0930: 70 03 00 04 22 05 00 02 BB 01 BA 04 00 04 77 04 :99
0940: 00 01 60 02 00 01 06 03 00 06 77 05 00 01 60 01 :51
0950: 46 01 60 01 46 01 60 01 46 01 60 01 46 01 00 01 :40
0960: 77 01 70 01 77 01 70 01 77 01 70 29 00 00 16 00 :F9
0970: 04 EE 01 E0 03 00 01 33 02 00 01 33 03 00 01 04 :48
0980: 04 BB 01 80 03 00 03 33 01 DD 03 00 01 70 01 00 :FC
0990: 02 33 01 00 01 07 02 00 01 02 01 22 01 20 01 00 :88
09A0: 01 22 01 20 03 00 03 BB 01 AA 03 00 01 07 01 77 :33
09B0: 02 00 01 77 01 70 03 00 01 60 02 00 01 06 02 00 :5A
09C0: 02 77 04 00 02 77 04 00 01 64 01 06 01 64 01 06 :D2
09D0: 01 64 01 06 01 64 01 06 02 07 01 77 01 07 01 77 :D9
09E0: 01 07 01 77 29 00 00 15 00 01 EE 01 E0 01 00 02 :91
09F0: EE 01 E0 02 00 01 03 01 33 02 00 01 33 01 30 02 :72

Sum : 93 4D BA B2 1D 4A 4E 46 27 89 09 0F 3B 1A 2C 06 :DF

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F iSum \* Ascii Dump \*
0A00: 00 01 44 01 46 04 BB 02 00 04 33 01 DD 01 00 01 :39
0A10: 00 01 07 01 00 01 03 02 03 01 30 01 00 01 70 :86
0A20: 00 02 22 01 20 01 00 01 02 01 22 02 00 04 BB 01 :2E
0A30: AA 01 A0 02 00 01 77 01 00 01 07 01 70 01 00 01 :41
0A40: 77 03 00 01 60 02 00 01 06 02 00 02 77 04 00 02 :65
0A50: 77 04 00 01 46 01 64 01 46 01 64 01 46 01 46 01 :B0
0A60: 46 01 64 08 77 2B 00 00 14 00 02 EE 01 E0 02 00 :39
0A70: 02 EE 02 00 01 03 01 30 02 00 01 03 01 30 01 00 :5F
0A80: 01 04 02 44 03 BB 01 BA 01 40 04 33 01 3D 01 DD :52
0A90: 01 D0 01 00 01 07 01 00 02 3D 02 03 01 00 01 70 :61
0AA0: 01 02 01 22 01 02 01 22 02 00 01 22 01 20 04 BB :51
0AB0: 01 BA 01 AA 01 A0 01 00 01 07 01 70 01 07 02 77 :02
0AC0: 01 70 01 07 01 70 01 06 01 66 01 06 02 00 01 60 :C2
0AD0: 01 66 01 60 01 77 06 00 01 77 04 00 01 46 01 64 :6E
0AE0: 01 46 01 64 01 46 01 64 01 46 01 64 08 77 28 00 :AB
0AF0: 00 14 00 02 EE 01 E0 01 0A 01 AA 01 00 01 0E 02 :AD

Sum : E7 BB 7B EC B0 C7 B6 79 4A B2 AB FC 1B 3E A2 4C :39

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F iSum \* Ascii Dump \*
0B00: 00 01 33 01 30 02 00 01 03 03 01 00 01 04 02 :A7
0B10: 44 01 33 BB 01 44 01 40 01 DD 03 33 01 3D 02 DD 01 :BA
0B20: 00 01 70 01 03 01 DD 01 3D 01 DD 01 3D 01 30 01 :D6
0B30: 07 01 02 01 20 01 00 01 22 02 00 01 02 01 20 01 :76
0B40: AA 03 BB 01 BA 02 AA 01 00 01 07 01 70 04 77 01 :C5
0B50: 07 01 70 01 60 01 00 01 66 02 00 01 66 01 00 01 :AC
0B60: 06 01 77 06 00 01 77 04 00 01 60 01 46 01 60 01 :0A
0B70: 46 01 60 01 46 01 60 01 46 01 70 01 77 01 70 01 :F1
0B80: 77 01 70 01 77 01 70 01 77 28 00 00 14 00 02 EE :75
0B90: 01 A0 02 AA 01 00 01 0E 01 E0 01 00 01 33 04 00 :E1
0BA0: 01 33 01 00 03 44 01 4B 01 BB 03 44 02 DD 02 33 :DF
0BB0: 03 DD 01 00 01 70 01 3D 01 DD 01 3D 01 DD 01 3D :CB
0BC0: 01 D3 01 07 01 22 02 00 01 22 01 22 01 22 02 00 :6A
0BD0: 02 AA 02 BB 03 AA 01 00 01 77 01 00 04 77 01 00 :0C
0BE0: 01 77 01 60 06 00 01 06 08 77 04 00 01 64 01 06 :D5
0BF0: 01 64 01 06 01 64 01 06 01 64 01 06 01 77 01 07 :C4

Sum : C9 7F DB E0 7E EF 16 AE 70 20 26 AF 2F 4B 85 96 :2E

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F iSum \* Ascii Dump \*
0C00: 01 77 01 07 01 77 01 07 01 77 01 77 01 07 2B :00
0C10: 00 02 EE 01 A0 02 AA 01 A0 01 0E 01 E0 01 00 01 :3A
0C20: 33 04 00 01 33 01 00 03 44 01 4B 04 44 01 02 57 :70
0C30: DD 01 33 03 00 01 D0 02 77 01 DD 01 70 01 D7 01 :07
0C40: 77 01 DD 01 77 01 22 02 00 01 02 01 20 02 00 01 :19
0C50: 22 01 A0 02 AA 01 BB 03 AA 01 A0 01 77 01 07 04 :67
0C60: 77 01 70 01 77 01 60 06 00 01 06 08 77 04 00 01 :52
0C70: 46 01 64 01 46 01 64 01 46 01 64 01 46 01 64 08 :B7
0C80: 77 2E 00 00 14 00 01 0E 01 E0 01 0A 02 AA 01 A0 :FB
0C90: 02 EE 01 00 01 33 04 00 01 33 01 00 04 44 01 BA :5E
0CA0: 03 44 01 0B 03 DD 01 CC 02 DD 01 D0 02 77 01 D7 :03
0CB0: 01 77 01 D7 01 77 01 D7 01 77 01 22 02 00 01 02 :40
0CC0: 01 20 02 00 01 22 01 A0 03 AA 01 44 02 AA 01 A0 :90
0CD0: 01 77 01 07 04 77 01 70 01 77 01 60 06 00 01 06 :52
0CE0: 01 77 06 00 01 77 04 00 01 46 01 64 01 46 01 64 :52
0CF0: 01 46 01 64 01 46 01 64 08 77 0B 00 01 04 01 :28

Sum : EB A7 EA 60 19 5C 2A AB 5E C3 55 1C 31 64 57 9D :3B

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F iSum \* Ascii Dump \*
0D00: 02 00 01 04 03 44 01 00 01 04 03 44 03 00 01 04 :A3
0D10: 01 40 08 00 00 14 00 01 0E 01 E0 01 00 02 AA 01 :FE
0D20: A0 02 EE 01 00 01 33 04 00 01 33 01 00 03 44 01 :46
0D30: BB 01 84 03 44 01 00 02 DD 02 CC 02 DD 01 33 01 :7A
0D40: 73 01 37 01 77 01 33 01 73 01 37 01 77 01 22 :02
0D50: 00 01 02 01 22 02 00 01 22 01 00 03 AA 02 44 02 :41
0D60: AA 01 77 01 00 04 77 01 00 00 77 01 60 06 00 01 :7F
0D70: 06 01 77 06 00 01 77 04 00 01 60 01 66 01 60 01 :0A
0D80: 46 01 60 01 46 01 60 01 46 01 70 01 77 01 70 01 :F1
0D90: 77 01 70 01 77 01 70 01 77 0B 00 01 04 01 04 02 :9C
0DA0: 00 01 04 03 44 01 40 01 04 03 44 03 00 01 04 01 :E2
0DB0: 40 00 00 00 15 00 01 E0 01 00 01 AA 01 A0 01 0E :9D
0DC0: 02 EE 01 00 01 33 01 30 02 00 01 03 01 30 01 00 :91
0DD0: 01 04 01 44 03 BB 02 44 01 40 01 00 01 DD 01 D3 :A9
0DE0: 03 CC 01 DD 01 03 01 DD 01 33 01 DD 01 33 01 D3 :A9
0DF0: 01 3D 01 D0 01 02 01 20 02 00 01 22 01 00 01 02 :5C

Sum : B5 50 AD 07 FC 5B 6B 63 49 BE AF 9F 2B F6 A1 D0 :B9

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F iSum \* Ascii Dump \*
0E00: 01 20 01 00 02 AA 01 A4 03 44 01 AA 01 07 01 70 :DE
0E10: 04 77 01 07 01 70 01 60 01 00 01 66 02 00 01 66 :26
0E20: 01 00 01 06 01 77 06 00 01 77 04 00 01 64 01 06 :6E
0E30: 01 64 01 06 01 64 01 06 01 64 01 06 01 77 01 07 :D4
0E40: 01 77 01 07 01 77 01 07 01 77 01 07 0B 00 02 44 :01
0E50: 02 00 01 04 01 44 01 00 02 44 01 04 03 44 01 40 :20
0E60: 02 00 02 44 0B 00 00 15 00 02 EE 02 00 01 0E 02 :68
0E70: EE 01 00 01 03 01 30 02 00 01 03 01 30 01 00 01 :5D
0E80: 04 01 46 03 BB 02 44 01 40 01 00 01 00 01 DD 01 :B3
0E90: DC 04 CC 01 00 01 3D 01 D3 01 DD 01 D3 01 D3 01 :80
0EA0: DD 01 00 01 02 01 22 02 00 01 22 01 20 01 22 01 :6E
0EB0: 20 01 00 01 0A 01 AA 01 AA 04 44 01 07 01 70 01 :3E
0EC0: 07 02 77 01 70 01 07 01 70 01 06 01 66 01 06 02 :E1
0ED0: 00 01 60 01 66 01 60 01 77 06 00 01 77 04 00 01 :24
0EE0: 46 01 64 01 46 01 64 01 46 01 64 01 46 01 64 08 :B7
0EF0: 77 0B 00 02 44 02 00 02 44 01 00 01 04 01 44 01 :5C

Sum : 9B B9 5A 6E 3C 5B 53 32 31 ED A7 2C DB 33 05 7A :E6

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F iSum \* Ascii Dump \*
0F00: 04 01 44 05 00 02 44 0B 00 00 15 00 01 0E 02 EE :B3
0F10: 01 00 01 0E 01 EE 02 00 01 03 01 33 02 00 01 33 :6F
0F20: 01 30 02 00 04 BB 01 BA 01 44 02 00 01 0D 01 DD :DA
0F30: 04 CC 03 00 01 D3 01 DD 02 3D 03 00 01 22 01 20 :0B
0F40: 01 00 01 02 02 22 02 00 01 0A 01 AA 04 44 02 00 :2A
0F50: 01 77 01 00 01 07 01 70 01 00 01 77 03 00 01 60 :CF
0F60: 02 00 01 06 02 00 02 77 04 00 02 77 04 00 01 46 :40
0F70: 01 64 01 46 01 64 01 46 01 64 01 46 01 64 08 77 :E8
0F80: 0A 00 01 04 02 44 02 00 01 44 01 40 02 00 01 44 :24
0F90: 01 64 01 44 05 00 02 44 01 40 0A 00 00 15 00 01 :F6
0FA0: 0E 04 EE 04 00 01 33 02 00 01 33 03 00 01 08 04 :B1
0FB0: BB 01 40 03 00 01 DD 03 CC 0B 00 01 02 01 22 01 :DE
0FC0: 00 01 02 01 22 01 20 03 00 01 AA 03 44 03 00 01 :40
0FD0: 07 01 77 02 00 01 77 01 70 03 00 01 60 02 00 01 :D1
0FE0: 06 02 00 02 77 04 00 02 77 04 00 01 60 01 46 01 :AB
0FF0: 60 01 46 01 60 01 46 01 60 01 46 01 60 01 77 01 :71

Sum : 50 E6 3D B6 0C 5B 3F 19 20 8B 4E 5B 19 03 FC B9 :DA

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F iSum \* Ascii Dump \*
1000: 00 01 77 01 70 01 77 02 70 0A 00 01 04 02 44 02 :9A
1010: 00 01 44 01 40 01 00 01 0A 01 44 01 04 01 44 05 :20
1020: 00 02 44 01 40 0A 00 00 16 00 03 EE 05 00 01 22 :D1
1030: 01 30 01 07 01 73 04 00 04 BB 04 00 01 DC 02 CC :DB
1040: 0D 00 04 22 04 00 01 44 02 44 05 00 04 77 06 00 :A6
1050: 01 60 02 00 01 66 03 00 01 77 04 00 01 77 05 00 :66
1060: 01 64 01 06 01 64 01 06 01 64 01 06 01 64 01 06 :B0
1070: 01 00 01 07 01 77 01 07 01 77 01 07 01 77 0B 00 :8C
1080: 01 44 01 40 01 44 02 00 01 44 01 40 01 00 01 04 :59
1090: 01 44 01 04 03 44 01 40 02 00 01 44 01 04 01 44 :B3
10A0: 0A 00 00 16 00 02 EE 06 00 01 03 02 33 01 30 05 :65
10B0: 00 02 BB 05 00 01 DD 01 CC 0F 02 22 02 05 00 01 :A5
10C0: 44 01 44 07 00 02 77 05 00 01 06 02 66 01 60 03 :41
10D0: 00 01 77 04 00 01 77 05 00 01 46 01 64 01 46 01 :ED
10E0: 64 01 46 01 64 01 46 01 64 02 00 04 77 0C 00 01 :46
10F0: 44 01 40 01 44 02 00 04 44 01 40 01 04 04 44 02 :A4

Sum : D9 B6 06 A1 A4 B1 B2 0A CA B5 E7 BD B1 C4 BC 61 :AC

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F iSum \* Ascii Dump \*
1100: 00 01 44 01 04 01 44 0A 00 00 0C 00 04 44 0F 00 :FC
1110: 02 33 06 00 01 44 01 BB 00 00 01 07 02 77 01 70 :3B
1120: 05 00 02 22 0E 00 02 77 05 00 01 66 01 60 01 06 :84
1130: 01 66 03 00 01 77 04 00 01 77 05 00 01 46 01 64 :0F
1140: 01 46 01 64 01 46 01 64 01 46 01 64 02 00 04 77 :B1
1150: 0B 00 01 04 01 44 01 00 01 44 01 40 01 00 04 44 :25
1160: 01 00 01 04 04 44 01 00 01 04 01 44 01 00 01 44 :DF
1170: 01 40 09 00 00 0C 00 01 46 02 66 01 64 04 00 01 :6F
1180: 0E 06 EE 01 E0 02 00 01 03 02 33 01 30 04 00 02 :55
1190: 44 02 BB 0B 00 01 07 01 70 01 03 01 30 01 07 01 :C3
11A0: 70 03 00 04 22 0C 00 04 77 03 00 01 06 01 00 02 :2C
11B0: 06 01 60 01 00 01 60 02 00 01 77 04 00 01 77 05 :CA
11C0: 00 01 60 01 06 01 60 01 06 01 06 01 06 01 60 01 :9A
11D0: 06 01 00 01 07 01 70 01 07 01 70 01 07 01 70 :0A
11E0: 00 01 04 01 44 01 00 01 44 01 40 01 00 03 44 01 :1A
11F0: 00 01 00 01 04 01 44 00 00 01 04 01 44 01 00 01 :DB

Sum : 24 30 CB A4 71 AA C9 B0 97 12 3D 61 27 72 AD F0 :D1

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F iSum \* Ascii Dump \*
1200: 44 01 40 09 00 00 0C 00 01 46 03 44 04 00 01 0E :3B
1210: 02 EE 01 E0 01 0E 02 EE 01 E0 02 00 01 33 01 20 :1B
1220: 01 02 01 33 03 00 01 04 02 44 02 BB 01 B0 02 00 :FE
1230: 06 33 02 00 01 70 01 00 02 33 01 00 01 07 02 00 :E6
1240: 01 02 01 22 01 02 01 20 01 22 01 20 02 00 06 BB :51
1250: 02 00 01 07 04 77 01 70 03 00 01 60 02 00 01 06 :63
1260: 02 00 03 77 02 00 03 77 04 00 01 64 01 06 01 64 :C0
1270: 01 06 01 64 01 06 01 64 01 06 02 07 01 77 01 07 :68
1280: 01 77 01 07 01 77 0A 00 04 44 01 40 01 00 01 44 :D1
1290: 01 04 02 44 01 00 01 04 01 44 04 00 01 04 04 44 :E7
12A0: 09 00 00 0C 00 01 46 01 40 06 00 01 33 02 EE 02 :A4
12B0: 00 02 EE 01 E0 02 00 01 33 02 00 01 33 03 00 03 :43
12C0: 44 03 BB 01 00 01 DD 01 D3 04 33 01 3D 01 D0 01 :20
12D0: 07 01 00 01 02 02 3D 01 30 01 00 01 70 01 00 01 :FC
12E0: 22 01 01 02 01 20 01 02 01 22 01 00 01 0A 01 :9A
12F0: AB 04 BB 01 BA 01 A0 01 00 02 77 02 00 02 77 02 :BC

Sum : 76 B3 D1 7C AE 7C 71 67 BC 5D DE 31 FD 75 53 FB :30



Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	+Sum	*	Ascii	Dump	*
1900:	77	01	FF	01	77	0E	00	00	0F	00	01	04	01	66	01	46	1BF		.	.	f.F
1910:	01	64	01	77	0E	02	EE	02	02	0E	01	02	EE	01	0E	02	00	3A	.	.	/F./
1920:	01	64	02	00	01	33	03	03	8B	03	44	01	00	01	00	01	00	181	.	.	7.D./
1930:	01	DC	04	CC	01	CD	01	00	02	00	03	03	D3	01	3D	03	00	65	.	.	7.7./
1940:	01	22	01	20	01	02	01	20	01	02	01	22	01	00	01	0A	9A		.	.	/
1950:	01	64	04	44	01	4A	01	A0	01	02	02	77	02	00	02	77	CE		.	.	D.J./
1960:	01	00	01	06	01	00	01	60	02	00	01	06	01	00	01	60	D5		.	.	/
1970:	02	77	04	00	02	77	04	00	01	46	01	64	01	46	01	64	52		.	.	F.d.F.d
1980:	01	46	01	64	01	46	01	64	08	77	0F	00	01	77	01	FF	5E		.	.	F.d.F.d
1990:	01	77	04	00	01	77	01	FF	01	77	0F	00	00	0E	00	01	BA		.	.	/
19A0:	04	01	46	01	64	01	04	01	66	01	44	01	0E	0E	01	61		.	.	F.d./	
19B0:	E0	01	0E	0E	EE	01	E0	02	00	01	33	01	30	01	03	01	2C		.	.	/
19C0:	33	03	00	01	08	02	8B	02	44	01	40	02	00	06	CC	0A	64		.	.	D.B./
19D0:	00	01	02	01	22	01	02	01	20	01	22	01	20	02	00	06	96		.	.	/
19E0:	44	02	00	01	07	04	77	01	70	03	00	01	60	02	00	01	A1		.	.	p./
19F0:	06	02	00	02	77	04	00	02	77	04	00	01	60	01	06	01	6B		.	.	/

```
Sum : E2 7B 6B A3 7E A9 27 4A D5 FC 05 13 2B 5C D0 AC :E9
```

Addr:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
1A00:	80	01	06	01	60	01	06	01	60	01	06	01	00	01	77	01	41	:	.	.	.
1A10:	70	01	01	70	01	70	01	70	01	11	00	01	77	01	FF	01	EC	:	.	.	.
1A20:	07	02	00	01	77	01	FF	01	77	10	00	00	00	00	01	04	8C	:	.	.	.
1A30:	01	66	01	40	01	00	01	46	01	64	01	0E	06	EE	01	E0	39	:	.	.	.
1A40:	02	00	01	03	02	33	01	30	04	00	02	BB	02	44	14	00	B7	:	.	.	.
1A50:	04	22	0C	00	04	77	03	00	01	06	01	00	01	00	01	60	20	:	.	.	.
1A60:	01	00	01	60	02	00	01	77	04	00	01	77	05	00	01	64	C2	:	.	.	.
1A70:	01	06	01	64	01	06	01	64	01	06	01	64	01	06	01	00	4C	:	.	.	.
1A80:	01	07	01	77	01	07	01	77	01	07	01	77	12	00	01	77	0A	:	.	.	.
1A90:	01	FF	02	77	01	FF	01	77	11	00	00	0E	00	01	04	01	16	:	.	.	.
1AA0:	44	02	00	01	04	01	44	0B	00	02	33	06	00	01	BB	01	93	:	.	.	.
1AB0:	44	16	00	02	22	0E	00	02	77	05	00	01	66	01	60	01	D3	:	.	.	.
1AC0:	06	01	66	03	00	01	77	04	00	01	77	05	00	01	46	01	B1	:	.	.	.
1AD0:	64	01	46	01	64	01	46	01	64	01	46	01	64	02	00	04	76	:	.	.	.
1AE0:	77	14	00	01	7F	02	FF	01	F7	12	00	00	16	00	02	EE	1C	:	.	.	.
1AF0:	07	00	02	33	06	00	02	44	05	00	01	3D	01	00	06	00	A2	:	.	.	.

```
Sum : C2 C6 CE 33 62 CC 17 99 3B B4 FE 75 B7 16 8D 17 :0A
```

Addr:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	#	Ascii	Dump	#
1B00:	01	07	02	77	01	70	05	00	02	22	05	00	01	BA	01	A0	:7C		.	W.p.	".....C.
1B01:	07	00	02	77	05	00	01	66	02	01	66	03	00	06	77	:D5		.	W.F.	".....F.	
1B02:	05	00	01	46	01	64	01	46	01	64	01	46	01	64	01	:46	50		.	F.d.F	".....d.F.d.F
1B03:	01	64	02	00	04	77	14	00	01	7F	02	FF	01	F7	12	:00	81		.	G.....F.	
1B04:	00	16	00	03	EE	05	00	03	02	33	01	30	04	00	03	:7D		.	.....	".....3.0.	
1B05:	44	01	BB	04	00	01	3D	01	DD	01	00	04	00	01	07	:1F	FE		.	D.7.=.	".....3.2.
1B06:	70	01	03	01	30	01	07	01	70	03	00	04	22	04	00	:1C		.	p.0.0.	".....p....."	
1B07:	BA	01	AA	01	AA	05	00	04	77	03	00	01	06	01	01	:92		.	C.x.	".....p....."	
1B08:	06	01	06	01	00	01	60	02	00	06	77	05	00	01	60	:1A	FF		.	.....W....."	
1B09:	00	01	06	01	06	01	06	01	06	01	60	01	06	01	00	:14		.	.....	".....p....."	
1BA0:	07	01	70	01	07	01	70	01	07	01	70	12	00	01	77	:1F	FF		.	p.p.p.	".....p....."
1BA1:	FF	02	77	01	FF	01	77	11	00	00	15	00	01	0E	04	:EE	17		.	W.W.	"...../"
1BA2:	04	00	01	03	01	30	01	03	01	30	03	00	01	04	03	:44	BD		.	.....0.....	".....0.....D
1BA3:	01	BB	01	B0	03	00	01	73	02	DD	01	00	03	00	01	:70	C8		.	9.....3	".....3....."
1BA4:	01	00	02	33	01	00	01	07	02	00	01	02	22	01	00	:68		.	.....3.....	"....."	
1BA5:	01	02	01	22	01	20	03	00	01	BB	02	AA	01	A0	03	:56		.	....."	".....9.x....."	

Sum : EF 46 C1 49 DB AB 0C 05 E0 DE 6F 49 6B F6 04 08 :B9

Addr:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	* Ascii	* Dump *
1C00:	01	07	04	77	01	70	03	00	01	60	01	06	01	60	01	06	:C7	..w..p..	..
1C10:	02	00	02	77	04	00	02	77	04	00	01	06	01	06	01	06	:C5	..w..p..w..	..
1E20:	01	06	01	60	01	66	01	60	01	06	01	00	01	07	01	70	:51	..p..p..	..w..p..
1C30:	01	07	01	70	01	07	01	70	11	00	01	77	01	FF	01	77	:F3	..p..p..p..	..w..w..
1C40:	02	00	01	77	01	FF	01	77	10	00	00	15	00	01	0E	02	:28	..w..w..	..
1C50:	EE	01	00	01	0E	01	EE	03	00	01	33	01	30	01	03	01	:5A	..p..p..	..3..0..
1C60:	30	03	00	04	44	02	B8	02	00	01	03	01	33	03	DD	01	:53	0..d..b..	..3..0..
1C70:	D0	01	00	01	07	01	00	01	03	01	3D	01	33	01	30	01	:82	..p..p..	..=..3..
1C80:	00	01	70	01	00	01	22	01	20	01	00	01	02	02	22	02	:E0	..p..p..	..="..
1C90:	00	01	08	01	B8	03	AA	01	AA	02	00	06	77	01	00	01	:97	..p..p..	..7..I..
1CA0:	06	01	00	01	60	02	00	01	66	01	00	01	60	02	77	04	:50	..w..w..	..
1CB0:	00	02	77	04	00	01	46	01	64	01	46	01	64	01	46	01	:1D	..w..w..	..F..d..F..d..F..
1CC0:	64	01	46	01	64	08	77	0F	00	01	77	01	FF	01	77	04	:92	d..F..d..w..	..w..w..w..
1CD0:	00	01	77	01	FF	01	77	0F	00	00	15	00	02	EE	02	00	:06	..w..w..	..
1CE0:	01	0E	02	EE	02	00	01	33	02	00	01	33	02	00	01	04	:72	..p..p..	..3..3..
1CF0:	03	44	01	4B	02	B8	01	80	01	00	01	03	01	33	01	3D	:78	D..K..9..	..=..3..

Sum : 63 72 BB 7D E3 4B B3 C9 57 6F 4B 35 DB 9A 7C 9F :8D

Addr:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
1D00:	04	DD	01	07	01	00	03	DD	01	33	01	00	71	70	01	02	1D3	.	.....	.3.....p.....	.
1D10:	01	22	02	09	01	22	01	20	31	22	01	20	01	00	01	08	1E4	.	.....	.....	.
1D20:	01	BB	01	BA	04	AA	01	07	02	77	02	00	02	77	01	70	192	.	7.....	.....w.....F.....	.
1D30:	01	60	01	66	04	00	01	66	01	06	01	77	0B	00	01	46	104	.	.....f.....f.....	.....w.....p.....	.
1D40:	01	64	01	64	01	64	01	46	01	64	01	46	01	64	0B	77	1E8	.	d.....f.....d.....f.....d.....w.....	.....f.....	.
1D50:	0E	00	01	77	01	FF	01	77	06	00	01	77	01	FF	01	77	1F4	.	.....w.....w.....w.....	.....f.....	.
1D60:	0E	00	00	15	00	01	E0	01	00	01	66	01	60	01	0E	02	1DE	.	.....=.....	.....f.....	.
1D70:	EE	02	00	01	33	02	00	01	33	02	00	01	04	03	44	01	1A9	.	.....3.....3.....	.....D.....	.
1D80:	4B	02	BB	01	B0	01	00	02	33	01	33	0D	0D	01	CC	01	1DB	.	K.....	.....3.....=.....f.....	.
1D90:	70	01	03	01	D3	01	00	DD	31	D3	01	D3	01	30	01	07	0172	.	p.....t.....	.....=.....t.....0.....	.
1DA0:	02	01	20	02	00	01	22	01	00	01	02	01	20	01	00	02	170	.	.....	.....	.
1DB0:	BB	01	BA	03	AA	01	44	01	07	01	77	01	70	01	07	01	162	.	7.....x.....D.....	.....w.....p.....	.
1DC0:	70	01	07	01	77	01	70	01	06	06	00	01	06	01	77	0B	152	.	p.....w.....p.....	.....=.....w.....	.
1DD0:	00	01	60	01	06	01	60	01	06	01	60	01	06	01	60	01	19A	.	.....	.....	.
1DE0:	06	01	70	01	07	01	70	01	07	01	70	01	07	01	70	01	1E3	.	p.....p.....	.....p.....p.....	.
1DF0:	07	0B	00	03	77	01	FF	01	F7	0B	00	01	7F	01	7F	03	10F	.	.....w.....	.....	.

Sum : 07 93 76 07 67 3A 6A 92 1A 4D C6 60 A4 56 7F C9 :83

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*	
1E00:	77	0B	00	00	14	00	01	0E	01	E0	01	00	02	66	01	60	150			W.....		
1E10:	02	EE	01	00	01	03	01	33	02	00	01	33	01	30	01	00	191			.....3.....f.....		
1E20:	02	BB	02	44	04	BB	01	00	03	33	02	DD	02	CC	01	70	117			7..D..7.....3..5..7..p		
1E30:	02	3D	01	DD	01	3D	01	DD	01	33	01	07	01	22	02	00	19A			..D..=..3.....3.....		
1E40:	01	02	01	22	02	00	01	22	01	00	03	BB	02	AA	02	44	1FC			.....7..x..D		
1E50:	02	77	01	00	02	77	01	00	02	77	01	06	06	00	01	60	1DB			W.....W.....W.....		
1E60:	01	77	0B	00	01	60	01	06	01	60	01	06	01	60	01	06	1BB			W.....W.....W.....		
1E70:	01	60	01	06	01	70	01	07	01	70	01	07	01	70	01	07	1DB			.....p.....p.....p.....		
1E80:	01	70	01	07	0B	00	00	01	7F	03	FF	01	F7	0B	00	01	17F	1B6			.....p.....7.....	
1E90:	03	FF	01	F7	0B	00	00	14	00	01	0E	01	00	06	02	112			.....f.....f.....f.....			
1EA0:	66	01	60	02	EE	01	00	03	01	30	02	00	01	03	01	1F4			f.....f.....f.....0.....			
1EB0:	30	01	00	03	BB	01	44	04	BB	01	03	03	33	01	DD	02	100			0.....7..D.....7.....3..D.....		
1EC0:	CC	01	CC	01	77	01	DD	01	70	D7	01	77	01	DD	01	190			7..W..W..3.....7..W..W..			
1ED0:	7D	01	77	01	22	02	00	01	02	01	20	02	00	01	22	01	164			.....W.....W.....		
1EE0:	0B	03	BB	01	AA	02	44	01	40	02	77	01	07	02	07	1F6			7..x..D.....@.....W.....W.....			
1EF0:	70	02	77	01	00	60	04	00	01	06	01	00	0B	77	04	1DA			p.....W.....W.....W.....			



Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum \* Ascii Dump \*
1F00: 00 01 46 01 64 01 46 01 64 01 46 01 64 01 46 01 4C ..F.d.F. d.F.d.F.
1F10: 64 08 77 0B 00 05 77 0B 00 05 77 0B 00 05 14 00 :18
1F20: 02 EE 01 06 02 66 01 60 01 0E 01 E0 01 00 01 03 :B5
1F30: 01 03 02 00 01 03 01 30 01 00 04 BB 01 44 03 BB :2B
1F40: 01 03 02 33 01 DD 03 CC 01 C0 01 77 01 D7 01 77 :6F
1F50: 01 D7 01 77 01 D7 01 77 01 D7 01 22 02 01 01 02 :A0
1F60: 01 20 02 00 01 22 01 08 02 BB 01 AA 03 44 01 40 :42
1F70: 02 77 01 07 02 77 01 07 02 77 01 00 01 60 04 :4A
1F80: 01 06 01 00 08 77 04 00 01 46 01 64 01 46 01 64 :E3
1F90: 01 46 01 64 01 46 01 64 08 77 28 00 00 14 00 02 :15
1FA0: EE 01 06 02 66 01 60 01 0E 01 E0 01 00 01 03 01 :54
1FB0: 33 02 00 01 33 01 30 01 00 04 BB 02 44 02 BB 02 :5F
1FC0: 33 02 DD 03 CC 01 30 01 00 01 73 01 37 01 33 01 :3B
1FD0: 73 01 37 01 77 01 37 01 22 02 00 01 22 01 20 02 :C6
1FE0: 00 01 22 02 BB 02 AA 03 44 01 00 02 77 01 00 02 :50
1FF0: 77 01 00 02 77 01 06 06 00 01 60 01 77 0B 00 01 :E3

Sum : AC EC 04 32 83 B0 E1 C8 5C A4 21 56 39 2B 77 E7 :B3

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum \* Ascii Dump \*
2000: 60 01 06 01 60 01 06 01 60 01 06 01 60 01 06 01 :A0
2010: 70 01 07 01 70 01 07 01 70 01 07 01 70 01 07 01 :0B
2020: 00 00 14 00 02 EE 01 E0 01 06 01 66 01 00 01 0E :53
2030: 03 00 01 33 02 00 01 33 02 00 01 0B 02 BB 01 B4 :ED
2040: 03 44 01 40 01 33 03 DD 01 DC 02 CC 01 00 01 00 :5A
2050: 01 33 01 DD 01 33 01 D3 01 3D 01 D3 01 00 01 02 :00
2060: 01 20 01 00 01 22 02 00 01 02 01 20 01 BB 03 AA :D4
2070: 01 A4 02 44 01 00 01 07 01 77 01 70 01 07 01 70 :56
2080: 01 07 01 77 01 70 01 60 06 00 01 06 01 77 0B 00 :E2
2090: 01 60 01 06 01 60 01 06 01 60 01 06 01 60 01 06 :A0
20A0: 01 70 01 07 01 70 01 07 01 70 01 07 01 70 01 07 :E4
20B0: 2B 00 00 14 00 02 EE 01 E0 02 00 02 EE 03 00 01 :03
20C0: 33 02 00 01 33 02 00 01 08 02 BB 01 B4 03 44 01 :31
20D0: 40 04 DD 01 DC 01 CC 01 C0 02 00 01 DD 01 3D 03 :03
20E0: D3 01 DD 01 00 01 02 01 22 01 02 01 22 02 00 01 :01
20F0: 22 01 20 04 AA 01 A4 01 44 01 40 01 00 01 07 02 :27

Sum : 6C 1C 04 35 94 BF 79 3E F0 72 14 BB 7B A0 AA 29 :EA

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum \* Ascii Dump \*
2100: 77 02 00 02 77 01 70 01 01 66 04 09 01 66 01 :97
2110: 06 01 77 0B 00 01 46 01 64 01 66 01 64 01 66 01 :29
2120: 64 01 46 01 64 0B 77 2B 00 00 15 00 01 01 EE 01 E0 :9C
2130: 01 00 02 EE 01 E0 03 00 01 33 01 01 03 01 03 01 :72
2140: 03 00 02 BB 04 44 02 00 01 0B 03 DD 01 CC 01 00 :86
2150: 03 00 03 D3 01 DD 03 00 02 22 01 20 01 00 01 02 :03
2160: 01 22 02 00 01 04 03 AA 01 44 01 40 02 00 06 77 :E2
2170: 01 00 01 06 01 00 01 60 02 00 01 06 01 00 01 60 :52
2180: 02 77 04 00 02 77 04 00 01 46 01 64 01 46 01 64 :52
2190: 01 46 01 64 01 46 01 64 0B 77 2B 00 00 16 00 04 :19
21A0: EE 01 E0 03 00 01 03 01 30 01 03 01 30 03 00 01 :40
21B0: 0B 01 BB 03 44 01 40 03 00 01 00 02 DD 01 CC 0B :17
21C0: 00 01 02 01 22 01 20 01 00 01 22 01 20 03 00 01 :90
21D0: 0A 02 AA 01 44 03 00 01 07 04 77 01 70 03 00 01 :F6
21E0: 60 01 06 01 60 01 06 02 00 02 77 04 00 02 77 04 :1B
21F0: 00 01 60 01 06 01 60 01 06 01 60 01 60 01 60 01 :9A

Sum : 50 EA 79 FE F6 DA 07 A1 11 6F 71 E6 0F 2B 5B 29 :BB

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum \* Ascii Dump \*
2200: 06 01 00 01 07 01 70 01 07 01 01 07 01 07 01 70 :29
2210: 00 00 17 00 03 EE 04 00 01 03 02 33 01 30 04 00 :7A
2220: 01 BB 03 44 05 00 01 0D 01 DD 01 DC 00 00 04 22 :03
2230: 05 00 01 0A 01 AA 01 A4 04 00 04 77 03 00 01 06 :E9
2240: 01 00 01 60 01 60 01 00 01 60 02 00 06 77 05 00 :4F
2250: 01 60 01 06 01 60 01 06 01 60 01 06 01 60 01 06 :A0
2260: 01 00 01 07 01 70 01 07 01 70 01 07 01 70 29 00 :95
2270: 00 18 00 02 EE 05 00 00 02 33 06 00 02 44 07 00 01 :96
2280: 0D 01 DC 0D 00 02 22 07 00 01 0A 01 77 05 00 02 :D9
2290: 77 05 00 01 66 02 00 01 66 03 00 06 77 05 00 01 :D2
22A0: 46 01 64 01 46 01 64 01 46 01 64 01 46 01 64 02 :B1
22B0: 00 04 77 2A 00 00 17 00 01 0E 01 64 01 46 01 64 :02
22C0: 06 00 02 44 06 00 01 33 01 DD 05 00 01 07 02 00 :E3
22D0: 01 70 05 00 02 22 06 00 01 BB 01 AA 06 00 02 77 :EA
22E0: 05 00 01 60 02 00 01 06 03 00 0A 77 05 00 01 60 :B5
22F0: 01 64 01 60 01 64 01 60 01 64 01 60 01 64 02 00 :B9

Sum : E6 13 DE FB BB FF 1F 63 F6 26 7F FF D7 F5 15 DE :DC

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum \* Ascii Dump \*
2300: 01 70 01 77 01 70 01 77 2A 00 00 17 00 02 EE 06 :09
2310: 00 02 33 05 00 04 44 04 00 01 03 01 33 01 DD 01 :9D
2320: 00 03 00 01 07 01 70 01 03 01 30 01 01 07 01 70 :03
2330: 00 04 22 04 00 01 0B 01 BB 01 AA 01 A0 04 00 04 :46
2340: 77 03 00 02 06 02 00 02 06 02 00 06 77 06 00 01 :6C
2350: 06 01 00 01 06 01 00 01 06 01 00 01 06 01 00 01 :20
2360: 07 01 00 01 07 01 00 01 07 2A 00 00 16 00 01 0E :68
2370: 02 EE 01 E0 05 00 02 33 04 00 01 04 04 44 01 40 :9D
2380: 03 00 02 33 02 DD 03 00 01 70 01 00 02 33 01 00 :C2
2390: 01 07 02 06 01 02 01 22 02 00 01 22 01 20 03 00 :79
23A0: 02 BB 02 AA 03 00 01 07 04 77 01 70 03 00 02 60 :C5
23B0: 02 06 01 00 02 77 04 00 02 77 04 00 01 60 01 06 :C6
23C0: 01 60 01 06 01 60 01 06 01 60 01 60 01 60 01 07 :41
23D0: 01 70 01 07 01 70 01 07 01 70 29 00 00 16 00 04 :A6
23E0: EE 05 00 02 33 04 00 01 B4 04 41 01 BB 02 00 01 :E8
23F0: 03 02 33 02 DD 01 D0 01 00 01 07 01 00 01 03 01 :F7

Sum : 52 0B 94 53 3A A5 9D EC 18 63 5A BF 34 1F 4B D1 :AC

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum \* Ascii Dump \*
2400: 33 01 D3 01 30 01 00 01 70 01 00 01 22 01 20 02 :F1
2410: 00 01 02 33 05 00 04 44 04 00 01 03 01 33 01 DD 01 :9D
2420: 00 06 77 01 00 02 06 01 00 01 06 01 60 01 00 02 :F2
2430: 60 02 77 04 00 02 77 04 00 01 46 01 64 01 46 01 :4E
2440: 64 01 46 01 64 01 46 01 64 0B 77 2B 00 00 15 00 :7B
2450: 01 0E 01 00 01 0B 01 B0 01 00 01 E0 04 00 02 33 :E8
2460: 03 00 01 0B 01 BB 03 44 01 4B 01 BB 01 B0 01 00 :CC
2470: 03 33 03 DD 01 00 01 07 01 00 01 33 03 03 01 :2B
2480: 01 70 01 02 01 22 03 00 02 22 01 20 01 03 03 BB :9E
2490: 03 AA 01 00 01 07 06 77 01 70 02 06 04 00 02 06 :12
24A0: 01 77 06 00 01 77 04 00 01 60 01 64 01 60 01 64 :B6
24B0: 01 60 01 64 01 60 01 64 01 70 01 77 01 70 01 77 :5E
24C0: 01 70 01 77 01 70 01 77 2B 00 00 15 00 01 EE 01 :FF
24D0: 00 02 BB 01 00 01 EE 04 00 02 33 03 00 01 0B 01 :F6
24E0: BB 01 B4 02 44 02 BB 01 B0 01 03 03 33 03 DD 01 :3F
24F0: D0 01 70 01 03 01 3D 01 D3 01 DD 01 3D 01 30 01 :A5

Sum : 90 B1 F7 D1 05 42 BD 5B 92 BE 99 72 0F 5D 2C DA :35

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum \* Ascii Dump \*
2500: 07 01 02 01 20 02 00 01 02 01 22 01 02 01 20 01 :18
2510: 0B 03 BB 03 AA 01 A0 01 07 02 01 22 01 22 01 20 01 :74
2520: 70 01 06 06 00 01 A0 01 07 02 01 22 01 22 01 20 01 :18
2530: 06 01 00 01 06 01 60 01 07 77 06 00 01 77 05 00 01 :DA
2540: 07 01 00 01 07 01 00 01 06 01 00 01 06 01 00 01 :20
2550: 14 00 01 0E 01 EE 01 08 02 BB 01 B0 01 EE 01 E0 :54
2560: 03 00 02 33 03 00 03 BB 02 44 01 4B 03 BB 04 33 :F7
2570: 04 DD 01 70 01 33 01 DD 01 44 01 4B 03 BB 04 33 :F7
2580: 01 22 03 00 01 22 01 20 01 00 01 22 04 BB 04 AA :F6
2590: 02 77 01 70 01 07 01 70 01 07 02 01 77 04 00 01 :4B
25A0: 04 00 01 06 01 00 01 77 06 00 01 77 04 00 01 60 :46
25B0: 01 06 01 60 01 06 01 60 01 06 01 60 01 06 01 60 :67
25C0: 01 07 01 70 01 07 01 70 01 70 01 70 01 70 28 00 :F8
25D0: 00 14 00 02 EE 04 BB 02 EE 03 00 02 33 03 00 03 :F1
25E0: BB 01 B4 04 BB 04 33 04 DD 01 07 01 D7 01 DD 01 :76
25F0: 77 01 7D 01 D7 01 DD 01 77 01 22 02 00 01 02 01 :4C

Sum : E5 A0 FF 0A 61 66 D5 B6 DD F6 3B C3 A1 7A AB FD :44

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum \* Ascii Dump \*
2600: 22 02 00 01 22 04 BB 04 AA 02 77 01 70 02 77 01 :1B
2610: 07 02 77 01 00 01 06 04 00 01 60 01 00 01 77 06 :6C
2620: 00 01 77 04 00 01 46 01 64 01 46 01 64 01 46 01 :1C
2630: 64 01 46 01 64 0B 77 2B 00 00 14 00 02 EE 04 BB :7A
2640: 02 EE 03 00 02 33 03 00 04 BB 01 4B 03 BB 04 DD :D5
2650: 04 CC 01 7D 01 77 01 7D 01 77 01 70 01 77 01 7D :30
2660: 01 77 01 22 02 00 01 22 01 20 02 00 01 22 04 AA :B4
2670: 04 44 02 77 01 70 02 77 01 07 02 77 01 00 01 06 :34
2680: 04 00 01 60 01 00 01 77 06 00 01 77 04 00 01 60 :C1
2690: 01 64 01 60 01 64 01 60 01 64 01 64 01 64 01 64 :12
26A0: 01 77 01 70 01 77 01 70 01 70 01 70 01 70 77 28 :5B
26B0: 00 14 00 01 0E 01 EE 01 08 02 BB 01 B0 01 EE 01 :7C
26C0: E0 03 00 02 33 03 00 03 BB 01 B4 01 44 03 BB 04 :95
26D0: DD 04 CC 01 73 01 77 01 73 01 37 01 33 01 77 01 :F2
26E0: 73 01 37 01 22 01 01 02 01 22 03 00 01 22 04 :1F
26F0: AA 04 44 02 77 01 70 01 07 01 70 01 07 02 77 01 :D7

Sum : 7B 76 B5 54 DC 0A 5D 95 5F 3E 72 90 10 29 25 AB :44

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum \* Ascii Dump \*
2700: 00 01 60 04 00 01 06 01 00 01 77 06 00 01 77 05 :4B
2710: 00 01 06 01 00 01 06 01 00 01 06 01 00 01 06 01 :20
2720: 00 01 07 01 00 01 07 01 00 01 07 01 00 01 07 2B :48
2730: 00 00 15 00 01 EE 01 00 02 BB 01 00 01 EE 04 00 :B4
2740: 02 33 03 00 01 08 02 BB 02 44 01 4B 01 BB 01 B0 :00
2750: 01 0D 03 DD 03 CC 01 C0 01 0D 01 3D 01 01 3D :DC
2760: 01 33 01 DD 01 33 01 D0 01 02 01 20 01 22 01 20 :7F
2770: 02 00 01 02 01 20 01 0A 03 AA 03 44 01 40 01 07 :6E
2780: 02 77 02 00 02 77 01 70 01 06 06 00 01 60 01 77 :4B
2790: 06 00 01 77 04 00 01 60 01 60 01 60 01 60 01 60 :B2
27A0: 01 06 01 60 01 06 01 70 01 07 01 70 01 07 01 70 :D2
27B0: 01 07 01 70 01 07 2B 00 00 15 00 01 0E 01 00 01 :CF
27C0: 0B 01 B0 01 00 01 E0 04 00 02 33 03 00 01 0B 01 :E7
27D0: BB 01 84 03 44 01 BB 01 B0 01 00 02 DD 03 CC 02 :D6
27E0: 00 01 DD 03 3D 01 D3 01 DD 01 00 01 02 02 22 03 :FB
27F0: 00 01 22 01 20 01 00 0A 03 44 01 40 01 07 01 07 :4B

Sum : D6 FE F2 11 B0 A3 B2 A1 43 EA 0A CD F5 56 BF 96 :F1

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum \* Ascii Dump \*
2800: 77 01 70 02 60 04 00 02 06 01 77 06 00 01 77 04 :50
2810: 00 01 46 01 64 01 46 01 64 01 64 01 64 01 64 01 :4C
2820: 64 0B 77 2B 00 00 16 00 04 EE 05 00 02 33 04 00 :51
2830: 01 BB 04 44 01 4B 02 00 01 0B 02 DD 03 00 01 02 :CE
2840: 03 00 01 DD 03 DD 03 00 01 22 01 20 02 00 01 02 :6D
2850: 01 22 02 00 01 0A 02 AA 02 44 01 40 02 00 06 77 :E2
2860: 01 00 02 06 01 00 01 06 01 64 01 00 02 06 02 77 :E2
2870: 04 00 02 77 04 00 01 60 01 64 01 60 01 64 01 60 :4E
2880: 01 64 01 60 01 64 01 00 01 77 01 70 01 77 01 70 :FE
2890: 01 77 01 70 01 77 2B 00 00 16 00 01 0E 02 EE 01 :9F
28A0: E0 05 00 02 33 04 00 01 04 04 44 01 40 03 00 02 :B1
28B0: DD 02 CC 08 00 01 02 01 22 02 00 01 22 01 20 03 :25
28C0: 00 02 AA 02 44 03 00 01 07 04 77 01 70 03 00 02 :EE
28D0: 60 02 06 02 00 02 77 04 00 02 77 05 00 01 06 01 :60
28E0: 00 01 06 01 00 01 06 01 00 01 06 01 00 01 07 01 :21
28F0: 00 01 07 01 00 01 07 2A 00 00 17 00 02 EE 06 00 :4B

Sum : 04 CF C3 AC 47 7E 14 45 A2 C1 1B 1E 52 35 EE BF :FD

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum \* Ascii Dump \*
2900: 02 33 05 00 04 44 04 00 01 DD 01 DD 01 CC 01 D0 :00
2910: 0C 00 04 22 04 00 01 0A 01 AA 01 44 01 40 04 00 :76
2920: 04 77 03 00 02 06 02 00 02 06 02 00 06 77 05 00 :6E
2930: 01 60 01 06 01 60 01 06 01 60 01 06 01 60 01 06 :A0
2940: 01 00 01 07 01 70 01 07 01 70 01 07 01 70 29 00 :95
2950: 00 17 00 01 0E 01 E0 06 00 02 33 06 00 02 44 06 :94
2960: 00 01 DD 01 CC 0E 00 02 22 06 00 01 AA 01 44 06 :D9
2970: 00 02 77 05 00 01 60 02 02 00 01 60 01 60 01 60 :4E
2980: 00 01 46 01 64 01 46 01 64 01 46 01 46 03 00 06 :5D
2990: 64 02 00 04 77 2A 00 00 18 00 02 EE 05 00 02 33 :4D
29A0: 06 00 02 44 07 00 01 33 01 3D 04 00 01 07 02 77 :4A
29B0: 01 70 05 00 02 22 07 01 BB 01 BA 05 00 02 77 :66
29C0: 05 00 01 66 02 00 01 66 03 00 01 77 04 00 01 77 :CC
29D0: 05 00 01 46 01 04 01 46 01 04 01 46 01 04 01 46 :3C
29E0: 01 04 02 00 01 77 01 07 01 77 01 77 01 77 2A 00 :17
29F0: 00 03 EE 04 00 01 03 02 33 01 30 04 00 04 44 05 :B0

Sum : BA 9E A1 2F CE F3 9D 0A DE 65 BF A9 52 6C C5 D2 :60

Adrs: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum \* Ascii Dump \*
2A00: 00 02 33 01 3D 03 00 01 07 01 70 01 03 01 30 01 :25
2A10: 07 01 70 03 00 04 22 05 00 02 BB 01 BA 04 00 04 :26
2A20: 77 03 00 01 06 01 00 01 60 01 06 01 00 01 60 02 :4E
2A30: 00 01 77 04 00 01 77 06 00 01 06 01 00 01 06 01 :0A
2A40: 00 01 06 01 00 01 06 01 00 01 07 01 00 01 07 01 :22
2A50: 00 01 07 2A 00 00 16 00 04 EE 01 E0 03 00 01 03 :22
2A60: 01 30 01 03 01 33 03 00 01 0B 01 B4 03 44 01 04 :B2
2A70: 03 00 03 33 01 DD 03 00 01 70 01 00 02 33 01 00 :C2
2A80: 01 07 02 00 01 02 01 22 02 00 01 22 01 20 03 00 :79
2A90: 03 BB 01 AA 03 00 01 07 04 77 01 70 03 00 01 60 :C4
2AA0: 01 06 01 60 01 06 02 00 02 77 01 70 03 00 01 60 :6B
2AB0: 01 60 01 00 01 60 01 00 01 60 01 60 01 60 01 60 :BB
2AC0: 01 70 01 00 01 70 01 00 01 60 01 60 01 60 01 60 :F0
2AD0: 00 15 00 01 EE 01 E0 01 00 02 EE 01 E0 03 00 01 :BB
2AE0: 33 01 03 01 03 01 30 03 00 02 BB 01 B4 04 42 :57
2AF0: 00 04 33 01 DD 01 D0 01 00 01 07 01 00 01 03 02 :F6

Sum : BC EB 94 7A F2 A1 3C 77 31 F9 2E 61 EE 19 B1 :B3



Addrst	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	* Ascii	Dump *
2000	01	30	01	00	01	70	01	00	01	22	01	20	02	00	01	1E	BE	t.O...p...	...
2001	01	22	02	00	04	BB	01	AA	01	AA	02	00	06	77	01	1B2		...	...
2002	00	01	06	01	00	01	60	02	00	01	06	01	00	01	60	02	04		...
2003	77	04	00	02	77	04	00	01	40	01	64	01	64	01	64	01	145	...	...
2004	01	64	01	64	01	64	01	64	01	77	01	77	01	77	01	1E		...	...
2005	70	01	77	01	70	01	77	28	00	00	14	00	02	EE	01	E0	DE	...	...
2006	02	00	02	EE	03	00	01	33	02	00	01	33	02	00	01	0B	6D	...	...
2007	02	BB	04	44	01	80	04	33	01	3D	01	DD	01	D0	01	00	1B5	...	...
2008	01	07	01	00	02	3D	02	03	01	00	01	70	01	02	01	22	1B8	...	...
2009	04	00	01	22	01	20	04	BB	01	BA	01	AA	01	AA	01	00	1F	...	...
200A	01	07	06	77	01	70	01	60	01	66	04	00	01	66	01	06	30	...	...
200B	01	77	06	00	01	77	04	00	01	46	01	46	01	46	01	04	92	...	...
200C	01	46	01	04	01	46	01	04	01	77	01	07	01	77	01	07	98	...	...
200D	01	77	01	07	01	77	01	07	2B	00	00	14	00	02	EE	01	2D	...	...
200E	01	0A	01	AA	01	00	01	0E	03	00	01	33	02	00	01	E0		...	...
200F	03	02	00	01	0B	02	BB	03	44	01	BB	01	80	01	DD	03	93	...	...

Sum : 1C 09 53 E0 E7 C0 33 91 DC 23 7C 51 BD 93 B5 29 :BD

Addrst	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	* Ascii	Dump *
2010	01	30	01	00	01	70	01	00	01	03	01	DD	01	DD	01	3D	1E3	...	...
2011	01	01	07	01	07	01	02	01	20	03	00	01	02	01	02	01	7F	...	...
2012	22	01	20	01	AA	02	BB	01	BA	02	AA	01	00	01	07	02	1E1	...	...
2013	77	01	70	01	07	02	77	01	70	01	60	06	00	01	06	01	49	...	...
2014	77	06	00	01	77	05	00	01	06	01	00	01	06	01	00	01	10B	...	...
2015	06	01	00	01	06	01	00	01	07	01	00	01	07	01	00	01	22	...	...
2016	07	01	00	01	07	2B	00	00	14	00	02	EE	01	0A	02	AA	1F3	...	...
2017	01	00	01	0E	01	E0	01	00	01	03	01	33	02	00	01	33	60	...	...
2018	01	30	01	00	03	BB	01	BA	01	44	03	BB	02	DD	02	33	8C	...	...
2019	03	DD	01	00	01	70	01	3D	01	DD	01	3D	01	DD	01	3D	CB	...	...
201A	01	D3	01	07	01	22	03	00	01	02	03	22	02	AA	02	BB	93	...	...
201B	03	AA	01	00	03	77	02	00	03	77	01	06	06	00	01	60	12	...	...
201C	01	77	06	00	01	77	04	00	01	60	01	60	01	60	01	00	1E	...	...
201D	01	60	01	00	01	60	01	00	01	70	01	00	01	70	01	00	1B	...	...
201E	01	70	01	00	01	70	29	00	00	14	00	02	EE	01	0A	02	1D	...	...
201F	AA	01	A0	01	0E	01	E0	01	00	01	03	01	30	02	00	01	74	...	...

Sum : E3 DE B7 1E 5C 21 4F FB C6 B9 3D 51 1B 47 61 72 :69

Addrst	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	* Ascii	Dump *
2020	03	01	30	01	00	03	BB	01	BA	04	BB	01	0D	02	DD	01	55	...	...
2021	33	03	DD	01	00	01	77	01	70	01	07	01	77	01	DD	01	09	...	...
2022	7D	01	D7	01	77	01	22	02	00	03	22	01	00	01	22	01	3C	...	...
2023	0A	02	AA	01	BB	03	AA	01	A0	02	77	01	70	01	07	01	83	...	...
2024	70	01	07	02	77	01	00	01	60	04	00	01	06	01	00	01	60	...	...
2025	77	06	00	01	77	04	00	01	40	01	64	01	64	01	64	01	46	...	...
2026	40	01	64	01	00	01	64	01	70	01	77	01	70	01	77	01	1E	...	...
2027	70	01	77	01	70	01	77	2B	00	00	14	00	01	0E	01	E0	FD	...	...
2028	01	0A	02	AA	01	A0	02	EE	01	00	01	03	01	30	02	00	80	...	...
2029	01	03	01	30	01	00	0A	0B	01	48	03	BB	01	DD	03	DD	ED	...	...
202A	01	CC	02	DD	01	DD	02	77	01	D7	01	77	01	D7	01	77	96	...	...
202B	01	DD	01	77	01	22	01	22	03	22	00	01	22	01	0A	0C	93	...	...
202C	03	AA	01	44	02	AA	01	A0	02	77	01	70	01	07	01	70	A2	...	...
202D	01	07	02	77	01	00	01	60	04	00	01	06	01	00	02	77	68	...	...
202E	04	00	02	77	04	00	01	46	01	04	01	46	01	04	01	46	60	...	...
202F	01	04	01	46	01	04	01	77	01	07	01	77	01	07	01	77	C9	...	...

Sum : 61 75 7C AF AC 4F E6 0D EF D6 25 6F B3 5E CB E9 :0D

																	Dump		
2030:	01	07	01	77	01	07	28	00	00	14	00	01	0E	01	E0	01	:85	...	W..f.....
2031:	00	02	AA	01	A0	02	EE	01	00	01	03	01	33	02	00	01	:79	...	.....3.....
2032:	33	01	30	01	00	03	BB	01	44	01	48	03	BB	01	00	03	:76	...	3.0...7.D.k.9...
2033:	DD	02	CC	DD	01	37	01	77	01	33	01	73	01	37	01	1B	:	0.7.7.7.W.3.9.7.	
2034:	77	01	33	01	73	03	22	01	20	03	00	01	22	01	00	03	:8F	...	W.3.s.s.....
2035:	AA	02	44	02	AA	03	77	02	00	03	77	01	06	06	00	01	:A0	...	I.D.x.W.....
2036:	60	02	77	04	00	02	77	05	00	01	06	01	00	01	06	01	:68	...	.....W.....
2037:	00	01	06	01	00	01	06	01	00	01	07	01	00	01	07	01	:22	...	.....
2038:	00	01	07	01	00	01	07	2B	00	00	15	00	01	E0	01	00	:30	...	.....3.....
2039:	01	AA	01	A0	01	0E	02	EE	02	00	01	33	02	00	01	33	:B7	...	x...../.....
203A:	02	00	01	0E	01	BB	03	44	02	BB	01	80	01	00	02	DD	:5F	...	.....7.D.9.....
203B:	01	DC	03	CC	DD	01	DD	01	03	01	DD	01	33	01	D3	01	:33	...	7.7.7.7.0.3.t.3.
203C:	01	D3	01	33	01	DD	01	02	22	01	20	03	00	01	01	02	:26	...	7.2.3.3.....
203D:	01	20	01	00	02	AA	01	44	03	44	01	AA	01	07	02	77	:E6	...	.....I.....D.I.....
203E:	01	70	01	07	02	77	01	00	01	60	06	00	01	60	01	00	:D2	...	.....I.....P.....
203F:	02	77	02	00	02	77	05	00	01	60	01	00	01	60	01	00	:BD	...	W..W.....
Sum:	9B	73	AC	35	A5	25	33	7F	E6	DD	26	EA	A2	2E	2E	CB	:04		

Sum : 9B 73 AC 35 A5 25 33 7F E6 DD 26 EA A2 2E 2E CB :04

Addrst	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	* Ascii	Dump *
2040	01	60	01	00	01	60	01	00	01	70	01	00	01	70	01	00	AB	.....	.....
2041	01	70	01	00	01	70	29	00	00	15	00	02	EE	02	00	01	14	.....	.....
2042	0E	02	EE	02	00	01	33	02	00	01	33	02	00	01	0B	04	7C	.....	.....
2043	04	02	BB	01	80	01	00	01	DD	01	DD	01	DC	04	CC	01	4D	.....	.....
2044	00	01	3D	01	D3	01	DD	01	DD	01	D3	01	DD	01	00	01	1E2	.....	.....
2045	02	01	22	04	00	01	22	01	20	01	00	01	0A	01	AA	01	125	.....	.....
2046	A4	04	44	01	07	06	77	01	70	01	60	01	66	04	00	01	1AF	.....	.....
2047	66	01	06	01	00	02	77	02	00	02	77	05	00	01	40	01	1A9	.....	.....
2048	64	01	00	01	64	01	40	01	64	01	40	01	64	01	70	01	5B	.....	.....
2049	77	01	70	01	77	01	70	01	77	01	70	01	77	2B	00	00	1C8	.....	.....
204A	15	00	01	0E	02	EE	01	00	01	0E	01	EE	03	00	01	33	14A	.....	.....
204B	01	30	01	03	01	33	03	00	04	41	04	01	BB	02	00	01	BE	.....	.....
204C	01	DD	01	DD	04	CC	03	00	01	DD	03	03	00	01	22	03		.....	.....
204D	01	20	02	00	01	02	01	22	02	00	01	0A	01	AA	04	44	49	.....	.....
204E	02	00	06	77	01	00	01	06	01	00	01	60	02	00	01	06	1F2	.....	.....
204F	01	00	01	60	02	00	04	77	06	00	01	46	01	04	01	46	178	.....	.....



Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
3700:	20	02	00	06	44	02	00	01	07	04	77	01	70	03	00	01	66	...	D...	W.W.p...	
3710:	60	02	00	01	06	02	00	02	77	04	00	02	77	3C	00	00	9D	...	...	W.W.W...	
3720:	14	00	01	0E	06	EE	01	E0	02	00	01	03	02	33	01	30	64	...	...	...	3.0
3730:	04	00	02	44	02	BB	14	00	04	22	00	00	04	77	03	00	CB	...	D...	...	
3740:	01	06	01	00	01	06	01	00	01	00	01	60	02	00	06	77	51	...	...	...	W
3750:	3D	00	00	1F	00	02	33	06	00	01	44	01	BB	16	00	02	B0	...	...	...	
3760:	22	0E	00	02	77	05	00	01	66	01	60	01	06	01	66	03	E7	...	...	...	
3770:	00	06	77	05	00	01	60	01	04	01	60	01	04	01	60	01	B0	...	...	...	
3780:	04	01	60	01	04	02	00	01	70	01	07	01	70	01	07	2A	B8	...	...	...	
3790:	00	00	16	00	02	EE	06	00	01	02	02	03	01	30	05	00	7B	...	...	...	
37A0:	02	BB	05	00	01	3D	01	DD	06	00	01	07	02	77	01	70	C9	...	...	...	
37B0:	05	00	02	22	05	00	01	BA	01	A0	07	00	02	77	05	00	0F	...	...	...	
37C0:	01	06	02	66	01	60	03	00	06	77	05	00	01	40	01	00	97	...	...	...	
37D0:	01	40	01	00	01	40	01	00	01	40	03	00	01	70	01	00	3A	...	...	...	
37E0:	01	70	2B	00	00	16	03	00	EE	05	00	01	33	01	30	01	0E	...	...	...	
37F0:	03	01	33	04	00	03	BB	01	44	04	00	01	3D	01	DD	01	5F	...	...	...	

Sum : 09 91 59 0C DB A1 70 DA A0 91 A2 A6 9B D2 F1 4A IE3

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
3800:	00	04	00	01	07	01	70	01	03	01	30	01	07	01	70	03	FE	...	...	...	
3810:	00	04	22	04	00	01	BA	01	AA	01	A0	05	00	04	77	04	B5	...	...	...	
3820:	00	01	60	02	00	01	06	03	00	06	77	3D	00	00	15	00	3C	...	...	...	
3830:	01	0E	04	EE	04	00	01	33	02	00	01	33	03	00	01	0E	7E	...	...	...	
3840:	03	BB	01	44	01	04	03	00	01	33	02	DD	01	DD	03	00	2E	...	...	...	
3850:	01	70	01	00	02	33	01	00	01	07	02	00	01	02	01	22	08	...	...	...	
3860:	02	00	01	22	01	20	03	00	01	BB	02	AA	01	A0	03	00	55	...	...	...	
3870:	01	07	04	77	01	70	03	00	01	60	02	00	01	06	02	00	63	...	...	...	
3880:	02	77	04	00	02	77	3C	00	00	15	00	01	0E	02	EE	01	47	...	...	...	
3890:	00	01	0E	01	EE	02	00	01	03	01	33	02	00	01	33	01	6F	...	...	...	
38A0:	30	02	00	04	BB	02	44	02	00	01	03	01	33	03	DD	01	52	...	...	...	
38B0:	00	01	00	01	07	01	00	01	03	01	30	01	33	01	30	01	B2	...	...	...	
38C0:	00	01	70	01	00	01	22	01	20	02	00	01	02	01	22	02	E0	...	...	...	
38D0:	00	01	08	01	BB	03	AA	01	A0	02	00	06	77	03	00	01	99	...	...	...	
38E0:	60	02	00	01	06	02	00	02	77	04	00	02	77	05	00	01	67	...	...	...	
38F0:	06	01	00	01	06	01	00	01	06	01	00	01	06	01	00	01	20	...	...	...	

Sum : 40 C9 1A DC B9 B9 87 41 F6 7E C3 0C 7B BE 56 3D B5

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
3900:	07	01	00	01	07	01	00	01	07	01	00	01	07	2B	00	00	4A	.....	.....	.....	(.....)
3910:	15	00	02	EE	02	00	01	0E	02	EE	01	00	01	03	01	30	3C	.....	.....	.....	.....
3920:	02	00	01	03	01	30	01	00	01	0B	03	BB	01	B4	02	44	FD	.....	.....	.....	.....
3930:	01	40	01	00	01	03	01	33	01	3D	04	DD	01	07	01	00	A2	.....	.....	.....	.....
3940:	03	DD	01	33	01	00	01	70	01	02	01	22	04	00	01	22	33	.....	.....	.....	.....
3950:	01	20	01	00	01	08	01	BB	01	BA	04	AA	01	07	06	77	DB	.....	.....	.....	.....
3960:	01	70	01	06	01	66	01	06	02	00	01	60	01	66	01	60	11	.....	.....	.....	.....
3970:	01	77	06	01	77	04	00	01	40	01	70	01	00	01	40	01	06	7E	.....	.....	.....
3980:	01	40	01	00	01	40	01	00	01	70	01	00	01	70	01	00	68	.....	.....	.....	.....
3990:	01	70	01	00	01	70	29	00	00	15	00	01	E0	01	00	01	04	.....	.....	.....	.....
39A0:	EE	01	E0	01	0E	02	EE	41	00	01	33	01	30	02	00	01	37	.....	.....	.....	.....
39B0:	05	01	33	01	00	01	04	03	BB	01	B4	02	44	01	40	01	3B	.....	.....	.....	.....
39C0:	00	02	33	01	20	03	DD	01	CC	01	70	01	03	03	D3	01	6A	.....	.....	.....	.....
39D0:	DD	01	3D	01	D3	01	30	01	07	01	02	01	22	01	20	03	72	.....	.....	.....	.....
39E0:	00	01	02	01	20	01	00	02	BB	01	BA	03	AA	01	44	01	90	.....	.....	.....	.....
39F0:	07	06	77	01	70	01	60	01	00	01	66	02	00	01	66	01	12B	.....	.....	.....	.....

Sum : FC 41 0B 31 BF D5 93 7C 5A BE B9 D0 35 0B EB 76 IE2

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
3A00:	00	01	06	01	77	06	00	01	77	3C	00	00	14	00	01	0E	5C	...	...	...	
3A10:	01	E0	01	02	EE	01	00	0E	02	EE	01	00	01	33	04	00	DC	...	...	...	
3A20:	01	33	01	00	02	44	02	BB	04	44	01	00	03	33	02	DD	96	...	...	...	
3A30:	02	CC	01	70	02	3D	01	DD	01	3D	01	DD	01	33	01	07	B4	...	...	...	
3A40:	03	22	01	20	03	00	01	22	01	00	03	BB	02	AA	02	44	1D	...	...	...	
3A50:	08	77	01	60	06	00	01	06	02	77	04	00	02	77	3C	00	1F	...	...	...	
3A60:	00	14	00	01	0E	01	E0	01	0E	02	EE	01	E0	02	EE	01	D5	...	...	...	
3A70:	00	01	33	04	00	01	33	01	00	03	44	01	BB	04	01	41	B9	...	...	...	
3A80:	03	03	33	01	DD	02	CC	01	CC	01	73	01	D7	01	77	01	6B	...	...	...	
3A90:	DD	01	7D	01	D7	01	77	01	37	01	22	01	00	03	22	02	2E	...	...	...	
3AA0:	00	01	22	01	08	03	BB	01	AA	02	44	01	40	03	77	01	9A	...	...	...	
3AB0:	70	01	07	03	77	01	60	06	00	01	06	02	77	04	00	02	DF	...	...	...	
3AC0:	77	05	00	01	06	01	00	01	06	01	00	01	06	01	00	01	95	...	...	...	
3AD0:	06	01	00	01	07	01	00	01	07	01	00	01	07	01	00	01	23	...	...	...	
3AE0:	07	2B	00	00	14	00	02	EE	01	0E	02	EE	01	E0	01	0E	22	...	...	...	
3AF0:	01	E0	01	00	01	32	04	00	01	33	01	00	04	44	01	BB	53	...	...	...	

Sum : E4 A2 18 FE EC B3 7D 9C 3F 6F 1E BF 58 F1 BA 09 8B

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*						
3B00:	02	44	01	02	02	33	01	DD	03	CC	01	E0	01	77	01	D7	3E		.	D	3	3	7	7	w	w	.
3B10:	01	77	01	D7	01	77	01	D7	01	77	01	D7	01	22	02	00	15		.	w	.	w	.	w	.	w	.
3B20:	03	22	01	00	01	22	01	0B	02	BB	01	AA	03	44	01	40	45		.	.	.	.	.	.	.	.	
3B30:	03	77	01	70	01	07	03	77	01	60	06	00	01	06	07	77	59		w	.	w	.	w	.	w	.	
3B40:	05	00	01	40	01	00	01	40	01	00	01	40	01	00	01	40	0C		.	.	.	.	.	.	.	.	
3B50:	01	00	01	70	01	00	01	70	01	00	01	70	01	00	01	70	C8		.	.	.	.	.	.	.	.	
3B60:	29	00	00	14	00	02	EE	01	0E	02	EE	01	00	01	0E	01	3D		.	.	.	.	.	.	.	.	
3B70:	E0	01	00	01	23	04	00	01	33	01	00	04	44	02	BB	02	55		.	.	.	.	.	.	.	.	
3B80:	44	02	33	02	DD	03	CC	01	00	01	77	01	33	01	73	01	49		D	3	3	7	7	w	w	.	
3B90:	37	01	77	01	33	01	73	01	77	01	22	03	00	01	02	03	FB		.	w	.	w	.	w	.	w	
3BA0:	22	02	BB	02	AA	03	44	01	00	08	77	01	60	01	0E	01	EE		.	.	.	.	.	.	.	.	
3BB0:	06	07	77	7D	00	00	14	00	02	EE	01	E0	01	0E	01	EE	84		.	.	.	.	.	.	.	.	
3BC0:	01	00	01	0E	02	00	01	33	01	30	02	00	01	03	01	33	B1		.	.	.	.	.	.	.	.	
3BD0:	01	00	01	04	02	44	01	48	03	BB	01	40	01	33	03	DD	AB		.	.	.	.	.	.	.	.	
3BE0:	01	0C	02	CC	01	00	01	4B	03	33	01	DD	01	33	01	D3	D4		.	7	7	.	.	.	.	.	
3BF0:	01	3D	01	33	01	D0	01	02	01	20	03	00	01	02	01	22	30		.	.	.	.	.	.	.	.	

# SMCと通信

SMC-70/777用

## MIDI インタフェース

西田明宏

SMC70/777はハードウェアがすっきりしていて拡張が容易なマシンです。SMC-777ではCPUバスを引き出すのがたいへんですが、SMC-70では拡張バスコネクタが電源と本体との間にあり、間にはさみこむ形で拡張することができるのはごぞんじの通りです。そこで今回はリクエストの多かったMIDIインタフェースを作ってみました。

### MIDI とは

MIDI とは、Musical Instrument Digital Interface の略で、要は電子楽器をデジタル信号でコントロールする規格である。MIDI インタフェースによって、MIDI 規格を採用しているシンセサイザやリズムマシン・コントローラなどをSMCから自由にコントロールすることができるわけだ。MSXにはすでにヤマハなどからMIDIインタフェースなどが発売され、数々のソフトウェアも発売されている。SMCユーザーも負けずにMIDIの世界に飛びこもう。

### MIDI インタフェースの製作

それではさっそく基板の制作に入ろう。図1が回路図である。この基板の主役はUARTである8251Aである。このICは80系のシリアルインタフェースICでSMCのRS-232Cカードにも用いられている。MIDIではこのICを非同期8ビット、スト

ップビット1ビットで用いる。転送レートは31.25Kbaudで一般のRS-232Cと比べてとても早い周波数を用いる。今回はSMCのクロック周波数4.028MHzを1/128分周して使っている。これだと正確な周波数である4MHzと異なるが、誤差は2%程度なので十分実用になる。MIDIケーブルから入ってきた信号はフォトカプラPC900によって電気的に絶縁され、8251Aの入力となる。MIDI THRU端子は入力された信号をそのまま外部へ出力する端子で、MIDI用機器を3台以上つなげるときに必要なものである。SMCとMIDI機器の2台のみを続ける場合は省略してもかまわない。

表1のパーツリストを示す。製作上の注意点をあげると、フォトカプラPC900は6pin DIPと変則なので、ソケットを使う場合は8pin DIPのソケットを流用すればよいこと、8251AはMOS ICなのでできるだけICソケットを使うこと、5pin DINソケットは基板じかづけのものとパネルにビス止めるタイプとがあるので好みに応じて用いること、TTLの中でLS93のみ電源ピンの位置が違うので注意することなどである。回路図には書いていないが、デジタル回路製作時の常識として、0.01μF程度のセラ

ミックコンデンサをパスコンとして、ICの電源ピン間に接続してほしい。プリント基板はどんなものでもよいが、今回はサンハヤトから出ているシャープX1用ユニバーサルボードを使った。この基板のようにICが使われることを前提に電源パターンがすでにプリントされている基板の方が配線が楽なようだ。

SMCに接続するには次のふたつの方法がある。SMC-70の中央のコネクタまたはSMC-777で拡張カードなしの場合はHIF3A 50DというコネクタとFAS-50-17というコネクタを、お店で50芯フラットケーブルに圧着してもらう。そしてMIDIインタフェース基板のFAP-50-07#2というコネクタと接続する。SMC-70の後部拡張コネクタおよびSM-C777の拡張カードを介して接続する場合はアンフェノール57F-30500というコネクタとFAS-50-17というコネクタをフラットケーブルで圧着してもらうこと。また、前者と後者では、コネクタの信号の物理的配置は同じなのだがピン番号の数が違うので注意して同じ信号線を接続すること。表2、3にそれぞれの場合のコネクタ形状と信号名を示す。

### MIDI コードの構成

さて、MIDIインタフェースができたところで、今度はMIDIコードについて簡単に説明しよう。

MIDIでは音楽の情報を1バイト単位でやりとりする。そしてデータの最上位ビットが1のときはステータスバイトといってコマンドを表し、0のときは各ステータスバイトに続くデータとして扱われる。つまり、データは常に0～127までの値をとるわけである。このステータスバイトで送るメッセージにはヴォイスメッセージとモードメッセージのふたつに大きく分けることができる。

1) ヴォイスメッセージ

●ノートオフ (8 nH)



図1 SMC-70/777 MIDI インタフェース

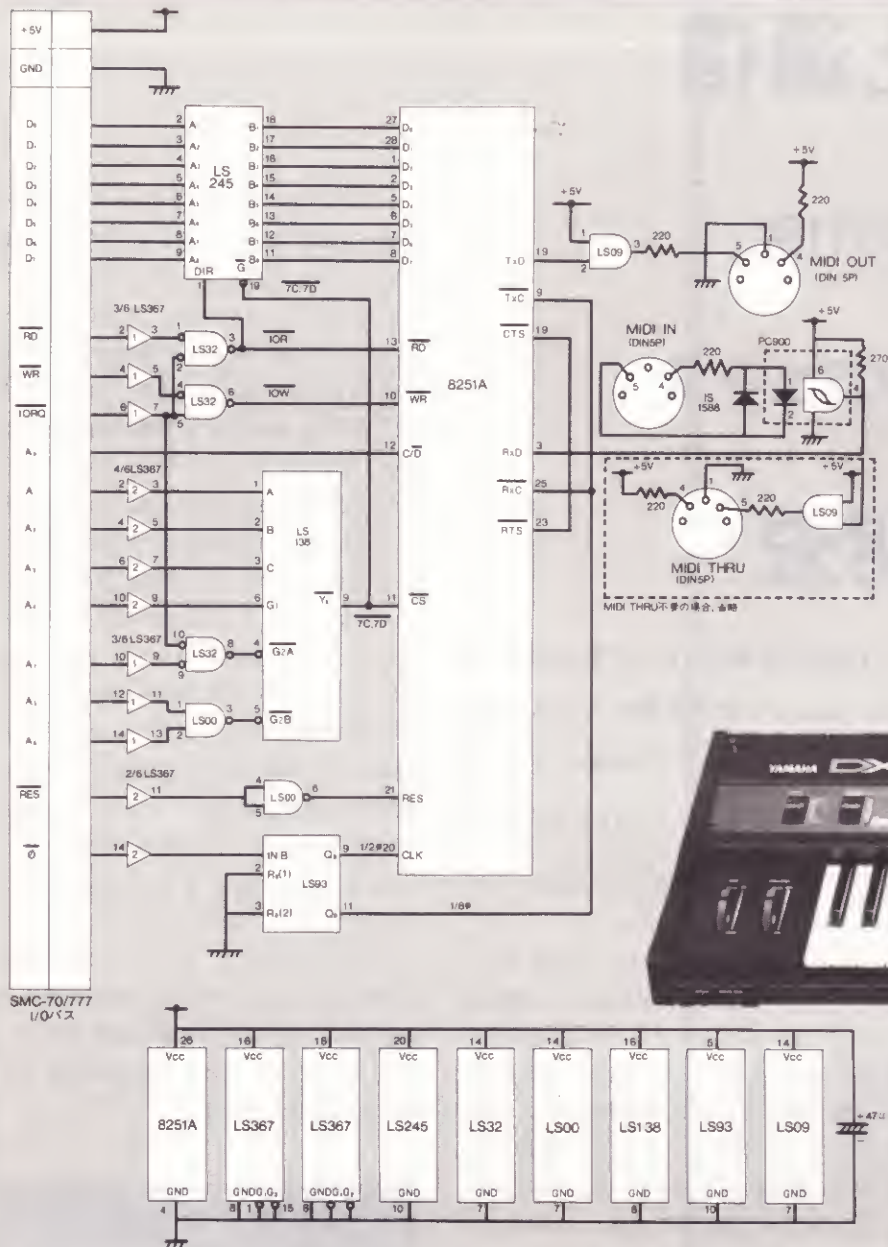


表 1

パーツ種別	型	名	数量	値段
IC	8251A		1	550
	LS367		2	85
	LS245		1	210
	LS138		1	75
	LS94		1	110
	LS00		1	35
	LS09		1	35
	LS32		1	45
フォトカプラ	PC900		1	175
ダイオード	IS1588		1	15 同等品で可
抵抗	220Ω		3(5)	10
	270Ω		1	
コンデンサ	47μ50v		1	20
	0.01μ		4-6	10 パスコン

コネクタ	FAP 50-03	1	500	
	FAS 50-17	1	600	
	HIF3A50D (アンフェノール 57F-30500)		700	どちらか 選択
			1000	
	DIN5P	2~3	150	基板用が 便利 何でも可
基板	ユニバーサル基板(X1用)	1	3650	
線材	ビニール線	適量		
ICソケット	28P	1	120	
	20P	1	80	
	16P	3	70	
	14P	4	70	
	8P	1	60	

秋葉原編へ

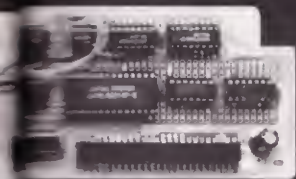


図 2

ピン配列  
50ピンコネクタ (DIPタイプ)



ピン No.	信号	信号の方向	定 格	ピン No.	信号	信号の方向	定 格
1	GND			2	+5V		
3	+5V			4	+5V	OUT	2,3,4ピンと共通
5	+5V	OUT	0.5A	6	+5V		
7	+5V			8	WAIT	IN	IIL -1.4mA
9	A8			10	A9		
11	A10			12	A11		
13	A12			14	A13		
15	A14	OUT	sink 22mA source -15mA	16	A15	OUT	sink 22mA source -15mA
17	A0			18	A1		
19	A2			20	A3		
21	A4			22	A5		
23	A6			24	A7		
25	GND			26	MI	OUT	sink 22mA source -15mA
27	GND			28	φ		
29	GND			30	GND		
31	WR	OUT	sink 22mA source -15mA	32	RD		
33	IORQ			34	HALT	OUT	sink 22mA source -15mA
35	INT	IN	IIL -1.4mA	36	RES		
37	D0			38	D1		
39	D2	IN/OUT	IN:IIL -1.7mA OUT:sink 22mA source -15mA	40	D3		
41	D4			42	D5	OUT/IN	IN:IIL -1.7mA OUT:sink 22mA source -15mA
43	D6			44	D7		
45	RTRN			46	RTRN		23ピンと共通
47	+12V	OUT	0.1A	48	+12V	OUT	24ピンと共通
49	-12V			50	NMI	IN	IIL -1.4mA

図 3

ピン配合  
50ピンコネクタ (アンフェノールタイプ)



ピン No.	信号	信号の方向	定 格	ピン No.	信号	信号の方向	定 格
1	GND			26	+5V		
2	+5V			27	+5V	OUT	2,3,4ピンと共通
3	+5V	OUT	0.5A	28	+5V		
4	+5V			29	WAIT	IN	IIL -1.4mA
5	A8			30	A9		
6	A10			31	A11		
7	A12			32	A13		
8	A14	OUT	sink 22mA source -15mA	33	A15	OUT	sink 22mA source -15mA
9	A0			35	A1		
10	A2			36	A3		
11	A4			37	A5		
12	A6			38	A7		
13	GND			39	MI	OUT	sink 22mA source -15mA
14	GND			40	φ		
15	GND			41	GND		
16	WR	OUT	sink 22mA source -15mA	42	RD		
17	IORQ			43	HALT	OUT	sink 22mA source -15mA
18	INT	IN	IIL -1.4mA	44	RES		
19	D0			45	D1		
20	D2	IN/OUT	IN:IIL -1.7mA OUT:sink 22mA source -15mA	46	D3	IN/OUT	IN:IIL -1.7mA OUT:sink 22mA source -15mA
21	D4			47	D5		
22	D6			48	D7		
23	RTRN			49	RTRN		23ピンと共通
24	+12V	OUT	0.1A	50	+12V	OUT	24ピンと共通
25	-12V				NMI	IN	IIL -1.4mA

音を消すというコマンドだ。nはチャンネル番号で、MIDIではこのチャンネル番号を各楽器に割り当てることによって最大16パートの別々の情報を送ることができる。このステータスバイトは2バイトデータを伴い、2バイトめが音の高さ、3バイトめが音の強さを表す。このノートオフメッセージは次のノートオンメッセージの音量0の場合で置きかえられるので、あまり使われない。

#### ●ノートオン (9nH)

音を出すためのメッセージである前のノートオフと同様2バイト目が音の高さ、3バイト目が音の強さを表す。最も多く使われるコマンドである。

#### ●ポリフォニック・キープレッシャー (AnH)

これはキーがどの程度の強さで押されているのかを示す情報で、微妙なタッチ・ビブラートやタッチ・ワウなどの効果をつけたいときに用いられる（そうだが）、現実にはあまり使われることはない。

#### ●コントロールチェンジ (8nH)

この部分はシンセサイザについているモジュレーションホイールやフットボリウムなどのコントローラの操作に関する情報をやりとりする部分である。この部分はシンセサイザによって機能がほとんど違うので、使用するシンセサイザのマニュアルなどを参照してほしい。また、このメッセージは2バイトのデータを判うが、2バイト目が7AH~7FHの場合はチャンネルモードメッセージというものに決められている。これについては後述する。

#### ●プログラム・チェンジ (CnH)

これはシンセサイザに音色の変更を伝えるコマンドである。1バイトのデータを伴い、0~127まで最大128通りの音色を指定することができる（もっとも実際はシンセサイザに備わっているプリセット機能の数によるわけだが）。

#### ●チャンネル・プレッシャー／アフター

指定したチャンネル全体の音量を指定するためのメッセージ。シンセサイザを複数制御するときに用いる。

#### ●ピッチホイール・チェンジ (EnH)

音程をずらすときに用いるメッセージ・ピッ



チベンダーなどを操作するときに用いる。

## 2) モード・メッセージ

先ほど述べたように、コントロールチェンジの中の特種なものにモードメッセージがある。

### ●ローカル・コントロール (BnH, 7AH)

2 バイト目に 7AH がくるとローカルコントロールメッセージとなる。3 バイト目に 0 がくると、ローカルコントロールがオフとなり、受信側のシンセサイザのキーを受け付けなくなる。3 バイト目が 7FH だともとのようにキーを操作することができる。

### ●オールノートオフ (BnH, 7BH)

3 バイト目に 0 が来ると、現在出ているすべての音を止める。

### ●オムニモード／ポリ・モノモード切替 (BnH 7CH～7FH)

オムニモードとはすべてのチャンネル番号のメッセージを受け入れるという意味で、受け入れた結果どうするかはそのキーボードに依存する。普通はシンセサイザ自身が必要な情報を取捨選択する。

モノモードは、複雑なヴォイスをそなえたシンセサイザのときに用いられる。モノモードでオムニモードを指定すると複数のヴォイスがそれぞれ異なるチャンネル番号に割り当てられるので 1 台で音色の異なる音を出すことができるようになる。この機能が使えるかどうかは手持ちのシンセサイザの取扱説明書を見てほしい。

ポリモードは通常の状態ではポリフォニックに音を出すモードである。

この各モードは以下のように指定する。

オムニモードオン: Bn 7D 00

オムニモードオフ: Bn 7C 00

モノモード : Bn 7E

[チャンネル番号]

ポリモード : Bn 7F 00

## 3) システムコモンメッセージ

F1～F7 の 7 つのコードで始まるメッセージはシステムコモンメッセージといい、MIDI システムで記録している曲の演奏に関するメッセージなど定義しているが、小規模なシステムについてはさほど使い道もないのでここでは説明は省略する。

## 4) システムリアルタイムメッセージ

F8～FF の 8 つのコードはシステムリアルタイムメッセージといい、各種のタイミング制御に用いられる(後続データはなし)。

### ●タイミングクロック (F8H)

4分音符 1 個につき 24 個の割合でデータが出る。これを使って MIDI システムの同期をとることができる

### ●スタート (FAH)

### ●コンティニュー (FBH)

### ●ストップ (FCH)

これらはシーケンサなどの制御に用いる。

### ●アクティブセンシング (FEH)

MIDI システムのチェック(回路の断線などに使います。普通はまず使うことはありません)。

### ●システムリセット

システム全体を初期状態にします。

## 5) システム・エクスクルーシブ

### メッセージ (F0H)

F0H はシステムエクスクルーシブメッセージで各メーカーのオリジナル機能を実現するために登録メーカーが自由に定義することのできるメッセージである。データの最初にメーカーの登録 ID 番号があらわれ、その後は任意のデータ列が来て、最後は F7H で終わることになっている。詳しいことは使う楽器の取扱説明書を見てほしい。

以上で MIDI のコマンドの簡単な説明を終わるが、もっと詳しく知りたい人のために参考書を紹介しておこう。

・デジタルキーボード操作術

立東社

・デジタルXブック

テストプログラムを作る。

## テストプログラムを作る

今月は MIDI を使ったラッサピアタの ORAN のようなプログラムをサンプルプログラムとして紹介する。また、この MIDI インターフェースのための基本的なサブルー

チンをリンクパッケージにしてみた。リスト 1 がリンクパッケージのソースリスト、リスト 2 がこのリンクパッケージを使ったデモプログラムである。このリストを走らせると、ラッサピアタの ORAN のように、SMC のキーボードによってシンセサイザを制御し音を出すことができる。リンクパッケージの使い方が、6 つのサブルーチンよりなっている。

MDINIT は MIDI インターフェースの初期設定を行うもので、他のルーチンと呼ぶ前に一度だけコールする必要がある。このルーチンによってチャンネルは 1 となる。CHSET はひとつの引数をもつ。これはチャンネルを変更するもので 1 チャンネルにするには 0、16 チャンネルにするには 15 を引数とすればよい。

ERRST は 8251 がオーバーラン、フレーミング、パリティなどのエラー状態となったときにこれをリセットするもので、実際にはあまり使う必要はないだろう。

DATOUT は 1 バイトのデータを送信するもので、後述する NOTEON, NTOFF 以外のメッセージを送りたいときに使う。

NOTEON はその名のとおり音を出すことを指示するサブルーチンである。ふたつの引数をもち、最初が音の高さ(中央のドが 60 でそれより小さいと低い音、大きいと高い音を表す)。音量は 64 が標準でそれより大きい音、小さいと小さい音、0 だと音を消すことを意味する。

NTOFF は音を消すためのコマンドでふたつの引数をもつ。引数の意味は NOTEON の場合と同じだが、2 番目の引数はタミーと考えてよい。

BASIC は遅いので、データを送り出すのは問題ないが、データを取り込むのは難しい。MIDI のボーレートは 31.25K とたいへん高いので、機械語で作ってもデータを取りこぼすことがある。

今回は動作確認用として簡単なプログラムを用意したが、近々音符入力式の MIDI 演奏プログラムを発表する予定である。その日のために、インターフェースを是非とも作っておいてほしい。

## リスト 1

File "midisub.asm

```

10 ;
20 ; SUBROUTINE PACKAGE FOR MIDI INTERFACE
30 ;
40 ;
50 ;
60 ;
70 ; ORG 100H
80 ; ENTRY MDINIT
90 ; ENTRY CHSET(BYT)
100 ; ENTRY ERRST
110 ; ENTRY DATOUT(BYT)
120 ; ENTRY NOTEON(BYT,BYT)
130 ; ENTRY NTTOFF(BYT,BYT)
140 ; ENTRY START
150 ;
160 ; MDSTAT EQU 7DH
170 ; MDDATA EQU 7CH
180 ;
190 ; Subroutine MDINIT
200 ;
210 ; INPUT NONE
220 ;
230 ; DESTROYS A,F
240 ;
250 ; MDINIT: A=A^A
260 ; (CHNUM)=A
270 ; P(MDSTAT)=A
280 ; P(MDSTAT)=A
290 ; P(MDSTAT)=A
300 ; A=40H
310 ; P(MDSTAT)=A
320 ; A=37H
330 ; P(MDSTAT)=A
340 ;
350 ;
360 ; SUBROUTINE CHSET
370 ;
380 ; INPUT (HL) = CHANNEL NUMBER
390 ;
400 ; DESTROYS A
410 ;
420 ; CHSET: A=(HL)
430 ; A=A&0FH
440 ; (CHNUM)=A
450 ;
460 ;
470 ; SUBROUTINE ERRST
480 ;
490 ; INPUT NONE
500 ;
510 ; DESTROYS A
520 ;
530 ; ERRST: A=37H
540 ; P(MDSTAT)=A
550 ;
560 ;
570 ; SUBROUTINE DATOUT
580 ;
590 ; INPUT (HL) = DATA
600 ;
610 ; DESTROYS A
620 ;
630 ; DATOUT: A=P(MDSTAT)
640 ; A=A&1
650 ; ->DATOUT,0
660 ; A=(HL)
670 ; P(MDDATA)=A
680 ;
690 ;
700 ; SUBROUTINE NOTEON
710 ;
720 ; INPUT (HL) = Key number
730 ; (DE) = Volume
740 ;
750 ; DESTROYS A,F
760 ;
770 ; NOTEON: B=BC
780 ; B=(HL)
790 ;
800 ; A=(CHNUM)
810 ; C=90H
820 ; A=A:C
830 ; (HL)=A
840 ; DATOUT

```

```

850 ;
860 ; A=B
870 ; A=A&7FH
880 ; (HL)=A
890 ; DATOUT
900 ;
910 ; A=(DE)
920 ; A=A&7FH
930 ; (HL)=A
940 ; DATOUT
950 ;
960 ; (HL)=B
970 ; BC=B
980 ;
990 ;
1000 ;
1010 ; SUBROUTINE NTTOFF
1020 ;
1030 ; INPUT (HL) = Key number
1040 ; (DE) = Volume
1050 ;
1060 ; DESTROYS A,F
1070 ;
1080 ; NTTOFF: B=BC
1090 ; B=(HL)
1100 ;
1110 ; A=(CHNUM)
1120 ; C=80H
1130 ; A=A:C
1140 ; (HL)=A
1150 ; DATOUT
1160 ;
1170 ; A=B
1180 ; A=A&7FH
1190 ; (HL)=A
1200 ; DATOUT
1210 ;
1220 ; A=(DE)
1230 ; A=A&7FH
1240 ; (HL)=A
1250 ; DATOUT
1260 ;
1270 ; (HL)=B
1280 ; BC=B
1290 ;
1300 ;
1310 ; WORK AREA
1320 ;
1330 ; CHNUM: DS 1
1340 ;
1350 ; END

```

## リスト 2

```

10
20 ; RASPIATER ORAN Simulation program by MIDI
30 ;
40 ; Oh! HITBIT No.6
50 ;
60 ; by A.Nishida 17-Mar-85
70 ;
80 ; Please Link MIDISUB.PAC
90 ;
100 ; CONSOLE 80:WIPE
110 ;
120 ; _GLOAD("RASPIATR.pic") ; アトリビュートファイルを読み込む
130 ;
140 ; NOTE$=" AZSXCXCVBNJMK,L, / ' * ' QW3E4RSTY7U8I0OP-[=]+<,>CHR$(10)
150 ; OLDNOTE=0
160 ; _MDINIT
170 ;
180 ; A$=INKEY$(1)
190 ; WHILE A$(<>CHR$(13))
200 ; IF A$="" THEN 260
210 ; IF A$>"a" AND A$<"z" THEN A$=CHR$(ASC(A$)-32)
220 ; _NOTEON(OLDNOTE,0)
230 ; NOTE=INSTR(NOTE$,A$)+54:IF NOTE=54 THEN 260
240 ; _NOTEON(NOTE,64)
250 ; OLDNOTE=NOTE
260 ; A$=INKEY$(1)
270 ; WEND
280 ;
290 ; _NOTEON(OLDNOTE,0)
300 ; END

```



# SMC SSTV プログラム



SMC 777による SSTV SYSTEM  
左側のビデオの上にあるのは  
スキャンコンバータ(自作)

## ●SSTVって何?

SSTVとはSlow Scan Televisionの略で、日本語にすると低速度走査テレビジョンということになります。音声回線を使用した画像伝送方式で、アマチュア無線では世界的に統一された規格があります。またプロの世界でも一部、同様な通信方法が使用されています。

国内、国外の友達と会話のできるアマチュア無線で、いつも交信に使用している機械を使って画像が交換できれば……、相手はどんな顔をしているんだろう？ 奥さんは美人かな？……そんな夢をこのSSTVが実現してくれました。現在アマチュア無線では、短波帯、またはアマチュア衛星を使った衛星通信で、多くのSSTVによる交信(お見合い?)が行われています(電話級でもOK!)。SSTVをやるには、Scan Converter (SSTV→一般のTV) という装置が必要ですが、ソフトウェアを作ればパソコンでも大丈夫です。アマチュア無線に限らず、音声を送れば画像を送ることも可能なので、いろいろな使用法が考えられます。たとえば、電話やワイヤレスマイクを使って画像

が送れるのです。送られてきた画像をカセットテープに録画(録音?)することも可能です。



DISKにSAVEしてあった画像を画面にLOADしたところ  
(「KEIKO」というFILEを選んだ)

## ●SSTVと普通のテレビの違いは?

SSTVというからにはテレビの一種なわけですが、一体どこがどう違うのでしょうか？ また、なぜ音声回線(電話など)で画像が送れるのでしょうか？

家庭のテレビの画面をよく見ると、画面がたくさん横線(走査線)から成っているのが分かると思います。家庭のテレビは、525本の走査線から1枚の画面が構成されていて、それが1秒間に30枚分送られてきます。このようにたくさんの画面情報を送るためには、広い幅の電波を必要とします。残念ながら、音声回線では送れません。走査線数を減らしたり、1秒間に送る画面の数を減らしていけば、必要な電波の幅が小さくなっていきます。音声回線(無声電話や有線電話)で画像を送るためには、走査線数や1秒間に送る画面数を大幅に減らさなければなりません。そこでSSTVでは、走査線数128本、1秒間に送る画面数1/8枚(つまり1枚の画面を送るのに8秒かかる)

という規格を用いています。ですから、家庭のテレビのように動く画像や細かい画像は、残念ながら望めません。しかし、相手の顔を鑑賞(?)したりするには十分なのです！ カラーSSTVは、アマチュア無線家の間で実験が進められているようですが、まだ統一規格はできあがっていないようです。



左側の画像に文字(Cブロック)を重ねて送信しそれを  
送り返してもらい受信したのが右側の画像(少々画像  
にノイズがはいっている)

## ●SMC-777でSSTVを!

SMC-777 CでSSTVの送受信ができるプログラムを作ってみました。友人の小林健一氏がSMC-70用に作ったプログラムをSMC-777用に改造し、DISK関係のプログラムを追加したものです。プログラムはBASICとマシン語で構成されています。

### 特徴

- 外付のハードウェアなしでSSTVの送受信が可能(ハンダごて不要!)
- SSTV信号の入出力はカセットインタフェースを使用。スピーカから送信音を出すことも可能。
- 受信した画像をフロッピーディスクにSAVE, LOAD可能。
- 受信した画像をプリンタで、ハードコピー可能。

#### 必要なシステム

- SMC-777+カラーパレット(SMC-777C)
- RGBモニタ

●カセットインタフェースケーブル(付属)  
今回はカメラ入力機能がありませんので、カメラから絵を送信することはできませんが、あらかじめBASIC、あるいはグラフィックスエディタなどで作成した絵をフロッピーディスクに入れておけば、いろんな絵を送信することができます。



#### ●プログラムの打ち込み方

プログラムはふたつ。ひとつはSSTV BASICプログラムで、もうひとつはSSTVマシン語LOADERプログラム(これもBASICで書いてある)です。SSTV用のディスクレットを1枚用意してください。必要なファイルは、BASIC.EXEとPIC.PACです。手順は以下のとおり。

- ①SSTV BASICプログラムを打ち込んで、“SS.BAS”というファイル名でSAVEします。
  - ②同様に、マシン語LOADERプログラムを“SS.MAC”というファイル名でSAVEします。
  - ③SMC-777の電源を入れ直して、BASICを再スタートさせます。
  - ④MCLEAR & HC000と入力します。
  - ⑤RUN “SS.MAC”と入力します。
- 約20秒後にSS.MACが終了し、SMC-777のメモリ(&HC000~&HC6FFF)にマシン語プログラムがLOADされます。

⑥次に、LOAD “SS.BAS”と入力し、BASICプログラムをLOADします。

⑦SAVE/C “SSTV.BAS”, &HC000, &HC6FFと入力すると、Enter Startup Command Line(s)?と聞いてきますので、LINK “PIC ⊙ RUN ⊙”と入力し、もう一度リターンキーを押します。

これでBASICプログラムとマシン語プログラムが、まとめて“SSTV.BAS”というファイル名でSAVEされました。以後プログラムをスタートさせるには、Sony FILERのMENUでBASICとSSTV.BASを選択するだけでOKです。



#### ●プログラムをRUNさせてみよう

プログラムをスタートさせると、図1のような画面が現れます。左右どちらのブロックに表示するかを指定することができます。ヘルプキーを押すと送信状態に切り替わり、受信ブロックに表示されている画像に送信ブロックの文字を重ね合わせたSSTV信号が、カセットインタフェースから得られます。また同時に、スピーカからもSSTV信号音が聞こえてきます。もう一度ヘルプキーを押すと受信状態に戻ります。HOMEキーを押すと、COPYモードに切り替わります。ディスクへのLOAD、SAVEおよびプリンタへのハードCOPYができます(プリンタはSMI-720かSMI-7020を使用)。ファイル名の指定などは、Sony FILERのMENUのように、カーソルパドルとリターンキーで簡単に行えます。また、ディス

クにSAVEされている画像を順次すべて表示させることも可能です(DISK→CRTモードでCLRキーを押す)。

送受信中のキーの機能は、表1のとおり。

#### ●カセットテープへの録音・再生

プログラムをスタートさせ、一度ヘルプキーを押して、送信状態にしてください。TXの文字が点滅して、送信状態を示しています。TXの文字の横の1Cは、受信ブロック1と送信ブロックCを重ね合わせたものを送信しているという意味です。

スピーカからはピロピロと音が聞こえていると思います。キーボードから何か文字を入力してみてください。文字が送信ブロックCに現れ、スピーカからの音が変わるのがわかります。では、このSSTV信号を録音してみましょう。

カセットテープにプログラムをSAVEするときと同じように、カセットインタフェースケーブルの赤いプラグをテープレコーダのMICジャックに入れ、録音します。スピーカからの音をマイクで録音してもよいでしょう。

キーボードから何かコメントを送信ブロックCに書いておきます。録音しながら、コメントをいろいろ書き変えてみてください。受信ブロックは真黒ですから、送信ブロックCの文字だけが、SSTV信号になっています。

テープを巻き戻して、今度は再生してみましょう。カセットインタフェースケーブルの白いプラグのテープレコーダのイヤホンジャックに入れ、再生状態にします。

キーボードの赤いヘルプキーを押して受信状態にすると、RXが点滅しています。横の2は、受信ブロック2に表示するという意味です。先ほど録音したSSTV信号が文字となって受信ブロック2に現れてきます。F5キーを押すと、RXの横の文字が2→&→1と変化します。&は受信ブロック1と2に交互に表示するという意味です(F5キーを押しても、表示が完了するまでは表



示位置は変わりません)。

### ●DISKへのSAVEとLOAD

HOMEキーを押してください。MENUが現れます。カーソルパドルでCRT→DISKを選んでリターンキーを押してください。受信ブロック1をSAVEするときは→SAVE,受信ブロック2をSAVEするときはSAVE←を選んでリターンキーを押してください。

FILENAME? と聞いてきますから、適当な名前を入れます。これで受信ブロックの絵がSAVEされました。カーソルをENDにもっていき、リターンキーを押すと、もとの画面に戻ります。

もう一度HOMEキーを押して、今度はDISK→CRTを選びます。LOADしたいFILEと受信ブロックをカーソルパドルで選んでください。HOMEキーを押すと元の画面に戻ります。

### ●BASICで送信用の

#### グラフィックを作ってみよう

BASICで、送信用の絵を作ってみましょう。サンプルプログラムを見てください。行番号100~140でグラフィック面にグラフィックを作成し、行番号150で“SAMPLE”というファイル名でディスクにSAVEしています。グラフィック面のX座標0~159, Y座標0~127にグラフィックを作成し、ディスクにSAVEします。GSAVEというコマンドを使用しますので、PIC.PACをLINKしておく必要があります (LINK “PIC.PAC” Ⓢ)。CIRCLE, BOX, LINE, PAINTなどのコマンドを使っていろいろなグラフィックを作ってみてください。

### ●セカンドフロッピーを使うには?

セカンドフロッピー (SMI-711) をお持ちの方は、Bドライブを画像DATA用に使ってみましょう。まず、SSTV用のディスクettをオリジナルディスクの“COPY”コマンドのGenerate System Disk で Dual Drive用に変更します。

次に、SSTVプログラムの最初の部分に90 LOG IN “B:” を追加します。これがBド

ライブにSAVE, LOADが可能になります。プログラムをスタートするときにA, B両ドライブにディスクettが入っていないとスタートできませんので注意してください。プログラム中でLOG IN “”やDISK RESETなどのコマンドを使えば、A, B両ドライブを使用して、多くのディスクettを使用

することができます。みなさんで実験してみてください。

#### 参考文献

●SMC-70対応SSTV送受信プログラム

小林健一 ハムジャーナルNO. 33

CQ出版社

●アマチュアのSSTV技術 CQ出版社

図1 画面表示

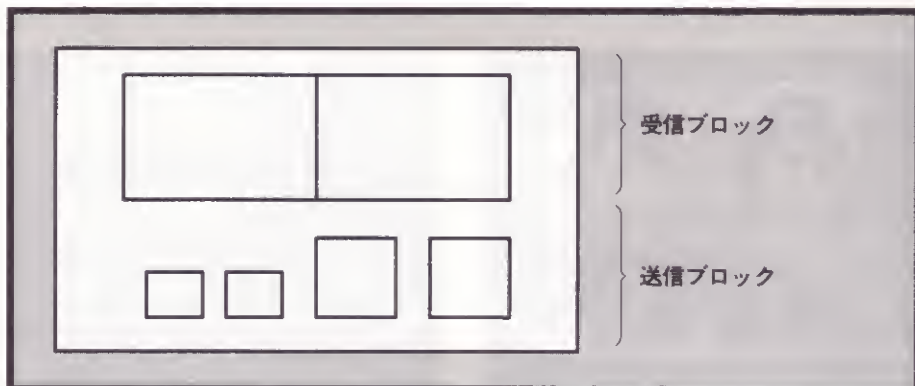


表1 送受信中のキーの機能

キ ー	機 能
SHIFT+ [F1] ~ [F5]	送信する送信ブロック A~D の指定
SHIFT+ [F5]	送信する受信ブロックの指定 [1] → [2]
[F1] ~ [F4]	文字を書き込む文字ブロックの指定 (カーソルの移動)
[F5]	受信時に使用する受信ブロックの指定 [1] → [2] → & [ ]
[方向キー]	カーソルの移動
[CLR]	その文字ブロックをすべてクリアーする
SHIFT+ [↑], [↓]	受信時 BRIGHT の増減
[HOME]	COPYモードへ
[ESC]	BRAKE する
[H]	送信↔受信 切りかえ

2

グラフィック作成  
サンプルプログラム

```

10 /
20 /
30 /
40 / Graphics program
50 /
60 /
70 /
80 /
90 /
100 FOR I=20 TO 120 STEP 20
110 FOR J=20 TO 120 STEP 20
120 CIRCLE (I,J),20,20
130 NEXT J
140 NEXT I
150 CALL GSAVE("sample",0,0,160,128)
160 END

```

```

100 /-----
110 /
120 / SSTV machine language program loader
130 /
140 / &HC000--&HC6FF Ver 2.0
150 /-----
151 / ----- HOW TO SAVE SSTV PROGRAMS -----
152 /
153 / First, save SSTV basic program and this program to DISK
154 /
160 / 1. start BASIC and enter ' MCLR ,&HC000 '
170 / 2. load this program and run
180 / 3. enter ' NEW ' ( erase this program )
190 / 4. load SSTV BASIC program
200 / 5. enter ' save/c "SSTV",&HC000,&HC6FF '
210 / 6. enter ' link"pic" , " run " , and return '
220 / 7. ---- SSTV programs were saved completely ----
230 /
240 / -----
1010 DEF INT A
1020 FOR A=&HC000 TO &HC6FF
1030 READ D$
1040 POKE A,VAL("&H"+D$)
1050 NEXT A
1060 END
2010 DATA C3, 09, C0, C3, C6, C4, C3, 85, C0, 3E, 13, D3, 1D, 3E, 04, D3,
2020 DATA 1D, DD, 21, 2D, C6, 11, 10, 00, 2E, 06, DD, 46, 02, DD, 4E, 03,
2030 DATA DD, 7E, 04, ED, 79, CB, D9, 3E, 08, ED, 79, DD, 19, 2D, C2, 1A,
2040 DATA C0, 1E, 02, CD, C8, C3, DD, 21, 6D, C6, DD, 22, 96, C6, DD, 36,
2050 DATA 08, 00, DD, 36, 09, 00, CD, 64, C4, 21, 2D, C6, CD, A1, C3, CD,
2060 DATA 7B, C3, 3E, 58, 32, B1, C6, 21, 08, 00, 01, 05, 54, 3E, 54, 1E,
2070 DATA 58, CD, 6A, C0, 01, 05, F4, 6C, 3E, 52, ED, 79, 04, 04, ED, 59,
2080 DATA 04, 04, 3E, 3A, ED, 79, CB, D9, 3E, 07, ED, 79, 05, 05, ED, 69,
2090 DATA 05, 05, ED, 69, C9, B7, 28, 58, 3D, 28, 45, 3D, 28, 20, 3D, 28,
2100 DATA 04, E1, 3E, 03, C9, DD, 7E, 02, FE, 01, 20, 12, 0A, 3D, FE, 04,
2110 DATA 30, EF, DD, E5, D5, 5F, CD, CB, C3, D1, E1, DD, E1, DD, 7E,
2120 DATA 01, FE, 01, 20, 1B, E5, 1A, 21, 2D, C6, FE, 01, 28, 0A, 21, 3D,
2130 DATA C6, FE, 02, 28, 03, E1, 18, C9, DD, E5, CD, A1, C3, DD, E1, E1,
2140 DATA 06, 00, DD, 7E, 00, FE, 01, C2, DC, C0, 7E, 47, 78, 32, AF, C6,
2150 DATA 3E, 80, 32, AE, C6, 3E, 14, D3, 1D, 21, 48, 08, CD, 5A, C0, C3,
2160 DATA 6B, C1, 08, CD, 25, C2, 08, CD, F8, C1, 78, 81, 47, D6, 66, D2,
2170 DATA 07, C1, 3E, 30, C3, F7, C0, 47, 1C, 3E, 9F, 8B, DA, 14, C1, 3E,
2180 DATA 01, C3, F2, C0, 3E, 2C, 0E, 6F, CD, F8, C1, 14, 3E, 7F, BA, DA,
2190 DATA 5D, C1, 2A, A1, C6, 01, 00, 10, 09, 01, A0, 80, 30, 01, 09, 22,
2200 DATA A1, C6, 2A, A5, C6, 01, 50, 00, 7A, DD, A6, 0C, 20, 01, 09, 22,
2210 DATA A5, C6, 0E, 6F, 3E, 1E, CD, F8, C1, CD, BC, C2, D8, 06, 08, 0E,
2220 DATA 6F, CD, F8, C1, 3E, 31, 10, F9, 3E, 2D, C3, DE, C1, 3A, AF, C6,
2230 DATA B7, CA, 6B, C1, 3D, 32, AF, C6, CA, 57, C0, 0E, 6F, 3E, 2A, CD,
2240 DATA F8, C1, CD, BC, C2, 06, 9D, 0E, 6F, CD, F8, C1, 3E, 31, 10, F9,
2250 DATA DD, 2A, 98, C6, DD, 46, 02, DD, 4E, 03, CB, D9, 3E, 08, ED, 79,
2260 DATA DD, 2A, 9A, C6, DD, 22, 98, C6, DD, 46, 02, DD, 4E, 03, CB, D9,
2270 DATA 3E, 48, ED, 79, DD, 6E, 00, DD, 66, 01, 22, A5, C6, DD, 2A, 9D,
2280 DATA C6, DD, 46, 02, DD, 4E, 03, CB, D9, 3E, 08, ED, 79, DD, 2A, 9F,
2290 DATA C6, DD, 22, 9D, C6, DD, 46, 02, DD, 4E, 03, CB, D9, 3E, 48, ED,
2300 DATA 79, DD, 6E, 00, DD, 66, 01, 22, A1, C6, 16, 00, 3E, 01, 2A, A1,
2310 DATA C6, 22, A3, C6, 2A, A5, C6, 22, A7, C6, 1E, 00, 06, 00, 0E, 6F,
2320 DATA CD, F8, C1, 3E, 05, C3, F2, C0, 3D, C2, F8, C1, 79, D6, 38, CB,
2330 DATA 3F, D2, 06, C2, FD, 09, 3D, 00, 00, 00, 00, C2, 06, C2, 3A, B0,
2340 DATA C6, EE, 10, 32, B0, C6, D3, 1D, FA, 20, C2, D6, 02, D3, 1D, C9,
2350 DATA D6, 00, D3, 1D, C9, C5, DD, 2A, 98, C6, 2A, A3, C6, 45, 4C, ED,

```



2360 DATA 78, CB, 43, C2, 3D, C2, 0F, 0F, 0F, 0F, C3, 42, C2, 23, 22, A3,  
 2370 DATA C6, 00, E6, 0F, 32, AB, C6, 7B, DD, A6, 0C, CA, 55, C2, 06, 12,  
 2380 DATA 10, FE, C3, 8E, C2, 2A, A7, C6, 45, 4C, 23, 23, 22, A7, C6, ED,  
 2390 DATA 78, FE, 20, CA, 6C, C2, 21, AC, C6, C3, 72, C2, 21, AB, C6, 21,  
 2400 DATA AB, C6, 22, A9, C6, 6F, 26, 02, 29, 29, 29, 7A, DD, CB, 0C, 66,  
 2410 DATA 28, 01, 1F, 1F, E6, 07, B5, 47, 4C, ED, 78, 32, AD, C6, 78, DD,  
 2420 DATA A6, 0D, CA, 9C, C2, 06, 04, 10, FE, C3, AB, C2, 3E, 00, 06, 0F,  
 2430 DATA 21, AD, C6, CB, 26, 30, 01, 78, 32, AC, C6, 2A, A9, C6, 7E, 21,  
 2440 DATA 1D, C6, 4F, 06, 00, 09, 7E, C6, 3A, C1, 4F, C9, DD, 2A, 96, C6,  
 2450 DATA 3A, B2, C6, B7, CA, CF, C2, 3D, 32, B2, C6, 2A, B3, C6, E9, DB,  
 2460 DATA 1B, 4F, E6, 01, C2, DA, C2, 3E, 28, C9, DB, 1A, 5F, E6, 60, C2,  
 2470 DATA E2, C3, 7B, CB, 71, CA, F2, C2, FE, 17, CA, 63, C3, FE, 1C, CA,  
 2480 DATA 6A, C3, FE, 0E, CA, 36, C4, FE, 08, CA, 28, C4, FE, 0D, CA, F6,  
 2490 DATA C3, FE, 19, CA, EA, C3, FE, 16, CA, 0D, C4, FE, 1C, CA, FA, C3,  
 2500 DATA FE, 17, CA, 19, C4, 1E, 00, FE, 01, CA, B3, C3, FE, 15, CA, CB,  
 2510 DATA C3, 1C, FE, 02, CA, B3, C3, FE, 18, CA, CB, C3, 1C, FE, 04, CA,  
 2520 DATA B3, C3, FE, 12, CA, CB, C3, 1C, FE, 06, CA, B3, C3, FE, 05, CA,  
 2530 DATA CB, C3, FE, 03, CA, 97, C3, FE, 08, CA, 71, C3, FE, 07, CA, AB,  
 2540 DATA C4, FE, 1D, CA, 99, C4, FE, 14, CA, C2, C4, FE, 1B, CA, B3, C4,  
 2550 DATA 3E, 05, C9, 21, B1, C6, 34, 3E, 1E, C9, 21, B1, C6, 35, 3E, 1E,  
 2560 DATA C9, 3A, 9C, C6, B7, FA, B9, C3, C2, B2, C3, 3E, 50, 2E, 32, C3,  
 2570 DATA BC, C3, 3E, 80, 2E, 26, C3, BC, C3, AF, 2E, 31, 32, 9C, C6, 01,  
 2580 DATA 05, FA, ED, 69, 3E, 05, C9, DD, 2A, 9F, C6, DD, 6E, 06, DD, 66,  
 2590 DATA 07, 22, 9F, C6, DD, 2A, 9F, C6, DD, 7E, 04, 01, 05, 5A, ED, 79,  
 2600 DATA 3E, 05, C9, 78, 07, 07, 07, 07, 4F, 06, 00, DD, 21, 4D, C6, DD,  
 2610 DATA 09, DD, 22, 96, C6, C3, 64, C4, 7B, C6, 41, 01, 05, 5C, ED, 79,  
 2620 DATA 7B, 07, 07, 07, 4F, 06, 00, 21, 4D, C6, 09, 22, 9A, C6, 3E,  
 2630 DATA 0A, C9, DD, 46, 06, DD, 4E, 07, ED, 59, DD, 34, 08, DD, 7E, 0A,  
 2640 DATA DD, BE, 08, F2, 64, C4, DD, 36, 08, 00, DD, 34, 09, DD, 7E, 0B,  
 2650 DATA DD, BE, 09, F2, 64, C4, DD, 36, 09, 00, C3, 64, C4, DD, 35, 08,  
 2660 DATA F2, 64, C4, DD, 7E, 0A, DD, 77, 08, DD, 35, 09, F2, 64, C4, DD,  
 2670 DATA 7E, 0B, DD, 77, 09, C3, 64, C4, CD, 0D, C4, DD, 46, 06, DD, 4E,  
 2680 DATA 07, 1E, 20, ED, 59, C9, DD, 36, 08, 00, DD, 36, 09, 00, DD, 7E,  
 2690 DATA 0B, 3C, 32, B2, C6, 21, 4E, C4, 22, B3, C6, C3, 64, C4, DD, 46,  
 2700 DATA 06, DD, 4E, 07, DD, 7E, 0A, 1E, 20, ED, 59, 04, 04, 3D, F2, 59,  
 2710 DATA C4, C3, FA, C3, DD, 6E, 00, DD, 66, 01, DD, 4E, 08, 06, 00, CB,  
 2720 DATA 21, 09, DD, 7E, 09, 07, 07, 07, 4F, 09, 07, 17, 4F, CB, 10,  
 2730 DATA 09, DD, 75, 06, DD, 74, 07, 0E, 19, 3E, 0E, D3, 18, ED, 61, 3E,  
 2740 DATA 0F, D3, 18, ED, 69, 3E, 0A, A7, C9, E1, 3A, AE, C6, EE, 80, 32,  
 2750 DATA AE, C6, F2, D0, C4, C3, E5, C0, 3A, B0, C6, EE, 80, 32, B0, C6,  
 2760 DATA 3E, 0A, C9, 3E, 04, D3, 1D, E1, E1, CD, 57, C0, 3E, 8E, 00, 00,  
 2770 DATA 00, C9, E1, C3, 57, C0, 3E, 00, 32, AE, C6, C0, 3E, AF, C6,  
 2780 DATA 3E, 04, D3, 1D, 21, 08, 48, CD, 5A, C0, 3A, 9C, C6, B7, F2, E6,  
 2790 DATA C4, EE, 50, 32, 9C, C6, E6, 50, 6F, 26, 80, 22, 92, C6, 16, 00,  
 2800 DATA 2A, 92, C6, 22, 94, C6, 01, 00, 10, 09, D2, 01, C5, 01, A0, 80,  
 2810 DATA 09, 22, 92, C6, CD, BC, C2, 21, 00, 00, CD, B2, C5, 06, 21, 10,  
 2820 DATA FE, CD, B2, C5, 06, 21, 10, FE, CD, B2, C5, 06, 21, 10, FE, CD,  
 2830 DATA B2, C5, 3E, 63, B9, F2, 4F, C5, 7A, A7, CA, 04, C5, 06, 00, 09,  
 2840 DATA 4D, 44, 21, A0, 0F, ED, 42, 69, 60, D2, 48, C5, 3E, 08, BA, DA,  
 2850 DATA DA, C4, 3A, 9C, C6, C3, E6, C4, 06, 1A, 10, FE, C3, 1F, C5, 06,  
 2860 DATA 00, 1E, 00, 0E, 00, 78, B1, 47, D6, 64, D2, 66, C5, 78, 06, 1C,  
 2870 DATA 10, FE, 47, C3, 90, C5, 47, C5, 3E, 9F, BB, 06, 1A, DA, BC, C5,  
 2880 DATA 3A, B1, C6, 91, 30, 01, AF, 2E, 1F, BD, 3B, 01, 7D, CB, 3F, 4F,  
 2890 DATA 06, 00, 21, 0D, C6, 09, 7E, CD, EA, C5, 06, 05, 10, FE, 1C, C1,  
 2900 DATA CD, B2, C5, 79, FE, 63, D2, A2, C5, 3E, AF, BB, D2, 55, C5, C3,  
 2910 DATA AB, C5, 3E, 78, BB, D2, 55, C5, 14, 3E, 7F, BA, D2, F0, C4, C3,  
 2920 DATA DA, C4, E5, C5, 21, C0, C5, E5, 2A, 8D, C6, 0E, 1D, 06, 2C, E9,  
 2930 DATA 3E, 42, 90, 21, 8F, C6, 4E, 77, B1, C1, E1, 4F, C9, 21, DB, C5,  
 2940 DATA 22, 8D, C6, ED, 78, F0, 10, FB, C3, E9, C5, 21, CD, C5, 22, 8D,  
 2950 DATA C6, ED, 78, F0, 10, FB, C3, E9, C5, C9, E5, 21, 90, C6, ED, 6F,  
 2960 DATA 7E, 2A, 94, C6, 45, 4C, CB, 43, CA, 03, C6, ED, 79, 23, 22, 94,  
 2970 DATA C6, E1, C9, E6, 0F, 07, 07, 07, 07, ED, 79, E1, C9, 00, 08, 01,  
 2980 DATA 09, 02, 0A, 03, 0B, 04, 0C, 05, 0D, 06, 0E, 07, 0F, 1E, 1A, 16,  
 2990 DATA 12, 0E, 0A, 06, 02, 1C, 1B, 14, 10, 0C, 0B, 04, 00, 00, 80, B0,  
 3000 DATA 04, 31, 00, 3D, C6, 32, D4, 39, EB, 2A, ED, 39, 24, 50, 80, DB,  
 3010 DATA 04, 32, 00, 2D, C6, C7, 39, C9, 25, EB, CD, E0, 35, 98, 06, 86,  
 3020 DATA 07, 41, 00, 98, 06, 00, 00, 04, 03, 1F, 03, F0, 3D, A6, 06, 94,  
 3030 DATA 07, 42, 00, A6, 06, 00, 00, 04, 03, 1F, 03, E6, 08, 74, 05, A2,  
 3040 DATA 07, 43, 00, 74, 05, 00, 00, 09, 07, 0F, 01, 39, 3C, 8C, 05, 8A,  
 3050 DATA 07, 44, 00, 8C, 05, 00, 00, 09, 07, 0F, 01, 2C, 3A, CD, C5, 16,  
 3060 DATA 00, 00, F0, D0, C0, 6D, C6, 7D, C6, 6D, C6, 50, 2D, C6, 2D,  
 3070 DATA C6, 40, C1, 90, B1, DC, 05, F0, 05, AB, C6, 00, 00, 00, 00, 00,  
 3080 DATA 07, 58, 00, 4E, C4, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,  
 3090 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,  
 3100 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,  
 3110 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,  
 3120 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,

```

100 '-----
110 '
120 '      SSTV program for
130 '
140 '      SMC-777C
150 '      or
160 '      SMC-777 + ColorPalette
170 '
180 '      (c)JA0RVJ,JR1WZI
190 '
200 '      10 May,1985      Ver 2.1
210 '-----
220 '
230 DIM FILE$(24),P$(16),BW$(15)
250 ON ERROR GOTO *ER
260 '
270 '---DEF PALETTE---
280 '
290 FOR I=0 TO 7
300 DEF PALETTE I,2*I,2*I,2*I
310 NEXT I
320 FOR I=8 TO 15
330 J=(I-8)*2+1
340 DEF PALETTE I,J,J,J
350 NEXT I
360 '
370 '-----
380 *SSTV
390 '
400 CONSOLE 40:CURSOR ON :CCOLOR ,0,0
410 CCLEAR 0,25 :PRINT (11,19);"(home)"
420 PRINT (12,18);"COPY"
430 SVINIT=&HC000
440 SVRX=&HC003
450 SVTX=&HC006
460 CALL SVINIT()
470 OUT &H1C,3
480 GCOLOR 9,0,9
490 LINE(0,128)-(319,128)
500 LINE(31,199)-(31,167)
510 LINE -(72,167)
520 LINE -(72,199)
530 LINE(87,199)-(87,167)
540 LINE -(128,167)
550 LINE -(128,199)
560 LINE(143,199)-(143,135)
570 LINE -(224,135)
580 LINE -(224,199)
590 LINE(239,199)-(239,135)
600 LINE -(319,135)
610 PAINT(0,129)
620 CALL SVRX()
630 '
640 '-----
650 *MENU
660 '
670 CONSOLE ,17,8:CCLEAR 17,8
680 CURSOR OFF
690 PRINT :PRINT
700 PRINT " --- SSTV UTILITyS --- ";

```

```

710 CPLOT(25,19),&H1:CPLOT(26,19),&H2
720 PRINT (27,19);" & return"
730 PRINT
740 PRINT "      COPY(DISK-->CRT)"
750 PRINT "      COPY(CRT-->DISK)"
760 PRINT "      COPY(CRT-->PRINTER)"
770 PRINT "      SSTV"
780 ACLEAR 17,8,,2
790 A=20
800 GOSUB 910
810 A$=INKEY$(1)
820 ACLEAR 17,8,,2,0
830 IF A$=CHR$(&H97) THEN A=A-1
840 IF A$=CHR$(&H9C) THEN A=A+1
850 IF A$=CHR$(&HD) THEN 960
860 IF A<20 THEN A=20
870 IF A>23 THEN A=23
880 GOSUB 910
890 GOTO 810
900 '
910 FOR I=3 TO 21
920 CPLOT(I,A),,,3,1
930 NEXT I
940 RETURN
950 '
960 IF A=20 THEN *DLOAD
970 IF A=21 THEN *DSAVE
980 IF A=22 THEN *HCOPI
990 IF A=23 THEN *SSTV
1000 GOTO 810
1010 '
1020 '-----
1030 *DLOAD
1040 CONSOLE:CCLEAR
1050 FOR I=0 TO 24
1060 FILE$(I)=" "
1070 NEXT I
1080 '
1090 DIR "*.PIC"
1100 FOR I=1 TO 24
1110 FOR J=11 TO 18
1120 F$=CHR$(CPOINT(J,I))
1130 FILE$(I)=FILE$(I)+F$
1140 NEXT J
1150 IF FILE$(I)=SPACE$(8) THEN 1180
1160 NEXT I
1170 '
1180 CCLEAR
1190 CONSOLE ,17,8
1200 CCOLOR ,2:CCLEAR 17,8
1210 '
1220 FOR I=0 TO 2
1230 FOR J=1 TO 8
1240 PRINT (9*I,J+16);FILE$(8*I+J);
1250 NEXT J:NEXT I
1260 '
1270 PRINT (29,17);"(home)-back"
1280 PRINT (30,18);"(c) r)-ALL"
1290 PRINT (30,20);"<- load"
1300 PRINT (30,21);"load ->"
1310 '
1320 *SFILE
1330 '
1340 X=1:Y=1:ACLEAR 17,8,,2,0
1350 GOSUB 1520
1360 A$=INKEY$(1)
1370 IF A$=CHR$(&H97) THEN Y=Y-1
1380 IF A$=CHR$(&H9C) THEN Y=Y+1
1390 IF A$=CHR$(&H96) THEN X=X-1
1400 IF A$=CHR$(&H99) THEN X=X+1
1410 IF A$=CHR$(&HD) THEN 1570
1420 IF A$=CHR$(&H94) GOTO *SSTV
1430 IF A$=CHR$(&H8E) GOTO *ALL
1440 IF X<1 THEN X=1
1450 IF X>3 THEN X=3
1460 IF Y<1 THEN Y=1
1470 IF Y>8 THEN Y=8

```



```

1480 ACLEAR 17,8,,2,0
1490 GOSUB 1520
1500 GOTO 1360
1510 '
1520 FOR I=(X-1)*9 TO (X-1)*9+7
1530 CPLOT (I,Y+16),,,3,1
1540 NEXT I
1550 RETURN
1560 '
1570 Z=1
1580 GOSUB 1700
1590 A$=INKEY$(1)
1600 GOSUB 1750
1610 IF A$=CHR$(&H97) THEN Z=Z-1
1620 IF A$=CHR$(&H9C) THEN Z=Z+1
1630 IF A$=CHR$(&HD) GOTO 1800
1640 IF A$=CHR$(&H94) GOTO *SFILE
1650 IF Z<1 THEN Z=1
1660 IF Z>2 THEN Z=2
1670 GOSUB 1700
1680 GOTO 1590
1690 '
1700 FOR I=1 TO 7
1710 CPLOT (29+I,Z+19),,,3,1
1720 NEXT I
1730 RETURN
1740 '
1750 FOR I=1 TO 7
1760 CPLOT (29+I,Z+19),,,2,0
1770 NEXT I
1780 RETURN
1790 '
1800 FILE$=FILE$(8*(X-1)+Y)
1810 IF FILE$=SPACE$(8) OR FILE$="" THEN *SFILE
1820 IF Z=1 CALL GLOAD(FILE$,0,0)
1830 IF Z=2 CALL GLOAD(FILE$,160,0)
1840 GOTO *SFILE
1850 '
1860 *ALL
1870 '
1880 ACLEAR 17,8,,2,0
1890 FOR K=1 TO 24
1900 X=INT(K/9)+1:Y=((K-8*(X-1)) MOD 9)
1910 GOSUB 1520
1920 IF FILE$(K)=SPACE$(8) THEN 1970
1930 IF (K MOD 2)=1 THEN XX=0 ELSE XX=160
1940 CALL GLOAD(FILE$(K),XX,0)
1950 ACLEAR 17,8,,2,0
1960 NEXT K
1970 GOTO *SFILE
1980 '
1990 '-----
2000 *DSAVE
2010 CCOLOR ,2,0:CCLEAR 17,8
2020 PRINT (3,18);" ->SAVE"
2030 PRINT (3,19);" SAVE<-"
2040 PRINT (3,20);" END "
2050 '
2060 Z=1:ACLEAR 18,3,,2,0
2070 GOSUB 2180
2080 A$=INKEY$(1)
2090 IF A$=CHR$(&H97) THEN Z=Z-1
2100 IF A$=CHR$(&H9C) THEN Z=Z+1
2110 IF A$=CHR$(&HD) GOTO 2230
2120 IF Z<1 THEN Z=1
2130 IF Z>3 THEN Z=3
2140 ACLEAR 18,3,,2,0
2150 GOSUB 2180
2160 GOTO 2080
2170 '
2180 FOR I=1 TO 6
2190 CPLOT (4+I,Z+17),,,3,1
2200 NEXT I
2210 RETURN
2220 '
2230 ON Z GOTO 2240,2250,*SSTV
2240 XX=0:GOTO 2260

```

```

2250 XX=160
2260 INPUT "FILE NAME?";F$
2270 CALL GSAVE(F$,XX,0,160,128)
2280 GOTO *DSAVE
2290 '
2300 '-----
2310 *HCOPY
2320 '
2330 CONSOLE 40,17,8,
2340 CCLEAR
2350 CCOLOR 7,2
2360 CCLEAR 17:LOCATE(0,17)
2370 PRINT "I will copy pictures to printer."
2380 PRINT
2390 PRINT " ----Set printer!----";
2400 PRINT " (home)->END"
2410 PRINT
2420 '
2430 *PIC
2440 PRINT "Select picture(1=left,2=right)";
2450 P$=INKEY$(1)
2460 IF P$="1" THEN P=0:GOTO *DAT
2470 IF P$="2" THEN P=160:GOTO *DAT
2480 IF P$=CHR$(&H94) GOTO *SSTV
2490 GOTO 2360
2500 '
2510 *DAT
2520 P$(0)=CHR$(&HF)+CHR$(&HF)+CHR$(&HF)+CHR$(&HF)
2530 P$(1)=CHR$(&H7)+CHR$(&HF)+CHR$(&HF)+CHR$(&HF)
2540 P$(2)=CHR$(&H7)+CHR$(&H7)+CHR$(&HF)+CHR$(&HF)
2550 P$(3)=CHR$(&H7)+CHR$(&H7)+CHR$(&H7)+CHR$(&HF)
2560 P$(4)=CHR$(&H7)+CHR$(&H7)+CHR$(&H7)+CHR$(&H7)
2570 P$(5)=CHR$(&H7)+CHR$(&H7)+CHR$(&H7)+CHR$(&H3)
2580 P$(6)=CHR$(&H7)+CHR$(&H7)+CHR$(&H7)+CHR$(&H1)
2590 P$(7)=CHR$(&H7)+CHR$(&H7)+CHR$(&H7)+CHR$(&H0)
2600 P$(8)=CHR$(&H7)+CHR$(&H7)+CHR$(&H6)+CHR$(&H0)
2610 P$(9)=CHR$(&H7)+CHR$(&H6)+CHR$(&H6)+CHR$(&H0)
2620 P$(10)=CHR$(&H6)+CHR$(&H6)+CHR$(&H6)+CHR$(&H0)
2630 P$(11)=CHR$(&H4)+CHR$(&H6)+CHR$(&H6)+CHR$(&H0)
2640 P$(12)=CHR$(&H0)+CHR$(&H6)+CHR$(&H6)+CHR$(&H0)
2650 P$(13)=CHR$(&H0)+CHR$(&H2)+CHR$(&H6)+CHR$(&H0)
2660 P$(14)=CHR$(&H0)+CHR$(&H2)+CHR$(&H2)+CHR$(&H0)
2670 P$(15)=CHR$(&H0)+CHR$(&H2)+CHR$(&H0)+CHR$(&H0)
2680 P$(16)=CHR$(&H0)+CHR$(&H0)+CHR$(&H0)+CHR$(&H0)
2690 '
2700 FOR I=0 TO 7
2710 BW$(I)=2*I
2720 NEXT I
2730 FOR I=8 TO 15
2740 BW$(I)=(I-8)*2+1
2750 NEXT I
2760 '
2770 PRINT @1,CHR$(&H1B);"T";"0B";
2780 FOR Y=0 TO 127
2790 PRINT @1,CHR$(&H1B);"S0640";
2800 FOR X=P TO 159+P
2810 GGET (X,Y)-(X,Y),G$(0)
2820 PRINT @1,P$(BW$(G$(0) AND &HF));
2830 NEXT X
2840 PRINT @1,
2850 NEXT Y
2860 PRINT:PRINT "COPY END..."
2870 INPUT "Continue(y/n)";C$
2880 IF C$="y" THEN GOTO *HCOPY
2890 IF C$="Y" THEN GOTO *HCOPY
2900 GOTO *SSTV
2910 '
2920 '-----
2930 *ER
2940 IF ERR=10 THEN *ER2
2950 *ER1
2960 PRINT (2,17);" ERROR!!! "
2970 PRINT (2,18);" No.=";ERR
2980 CLEAR:PAUSE 10:GOTO 100
2990 *ER2
3000 PRINT (2,17);"FILE NOT FOUND"
3010 CLEAR:PAUSE 10:GOTO 100

```

# 私の777

沖縄県 普久原朝宏

最近パソコンによるデータ通信が盛んになっていますが、私もデータ通信をハムが777用に開発したSSTVプログラムを使って楽しんでいますので、紹介したいと思います。ハムの分野では、パソコンはRTTY, SSTV, 交信局リストなどで使われているようです。RTTYとは、ラジオテレタイプのことです。パソコンのキーボードで入力したメッセージを、マークとスペースの信号に変換して、電波で世界各国のハム局の受信システムのブラウン管に表示させ、交信する方法です。新聞社で使用されているようなカタカタ音が出るテレタイプではなく、パソコンですので大変静かです。

SSTVとは、スロースキャンテレビジョンのことです。グラフィック、写真などの静止画像の送受信ができます。これまでは、残像管を使用してビデオカメラで紙に書いたメッセージや写真を撮影して送っていましたが、いまはパソコンを使用している局が増えてきているようです。SSTVは、1枚の画面を約8秒間要して完成させますので、8秒おきにどんどん絵を送り出せます。日本国内はもちろん外国局から送ってきた絵をディスクにセーブしたり、その絵に文字をインサートして送り返したりするのも可能です。画像はカセットテープにもセーブできます。

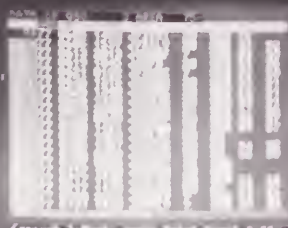
SSTVは主に白黒ですが、現在はカラー化が盛んになっており、私も現在準備中です。私の場合は2ドライブで、AドライブにSSTVシステムディスクを、Bドライブに画

写真 2



南アフリカからの画像。空中状態が悪く少しノイズがある。

写真 3  
MEMO



像データディスクを作成し、運用しています。写真だと約15枚くらいセーブできます。文字だけのメッセージだと、かなりの量のデータがセーブできそうです。

また、Memoも大変重宝しています。私は交信局リストに使用していますが、便利ですね。現在のところ、400局くらいしか入力してませんが、これまでの交信局を全部入力するには、まだまだ時間がかかりそうです。あと3,000局くらいかな！

無線機のダイヤルを回すと、国内国外の局が聞こえてきますが、聞こえている局がすでに交信済の局かどうかは、月日が経つとなかなか思い出せません。そこでMemoが活躍します。コールサインをインプットしてサーチすると、交信した局だと画面に出てきます。年度月日、周波数、時間などすぐ分かります。

写真 1 私のシステム

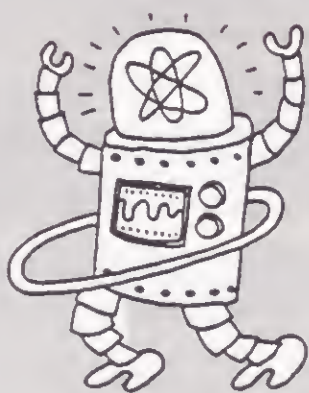




# ぽち2号のハード便り

## SMC用8/5.25インチディスクインタフェース

### ぽち2号



SMCシリーズはオペレーティング・システムをベースとしたコンピュータであり、CP/Mマシンとして高いパフォーマンスを誇っています。しかし、3.5インチディスクは、わずか280Kバイトの容量しかなく、CP/M上でプログラムを開発しているとすぐにディスクがパンクしてしまいます。だったら大容量ディスクをSMCにつないじゃえっ！という短絡的な発想から実現したのが今回発表する8/5.25インチディスクインタフェースです。

#### 基本方針

システムのハードウェアを拡張するときには、そのサポート・ソフトウェアが常に問題になります。しかも、今回のようにディスクを増設したりというようなオペレーティング・システム自体にかかわる場合は、そのオペレーティング・システムの一部を変更せざるを得なくなります。ところで、CP/Mの場合には各種のハードウェアの上で走るために、ハードウェアに依存する部分（ディスク入出力、コンソール入出力など）はBIOSと呼び、ユーザーが自分で作成することになっています。つまり、BIOSさえちゃんと作っておけば、8あるいは5.25インチディスクをCP/Mで直接利用できるわけです。とはいっても、やはりBIOSの作成(変更といった方が正しい)は大変なわけで、3.5インチのリード/ライ

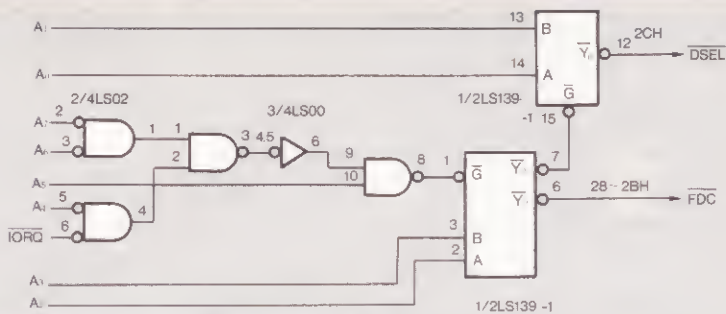
トルーチンを利用できれば最高。

ところがTINY BIOSのリストと777のディスクインタフェースの回路図を見比べていたところ、何のことはない、3.5インチの回路にちょっと手を加えればよいことが判明しました。そうすればそれほど大きなBIOSの変更なしに8/5.25インチディスクが接続できます。それで777回路図のディスクインタフェースを基本とすることになりました。また、I/Oポートはフロッピーディスク予約ポートである28H~2CHを利用します。

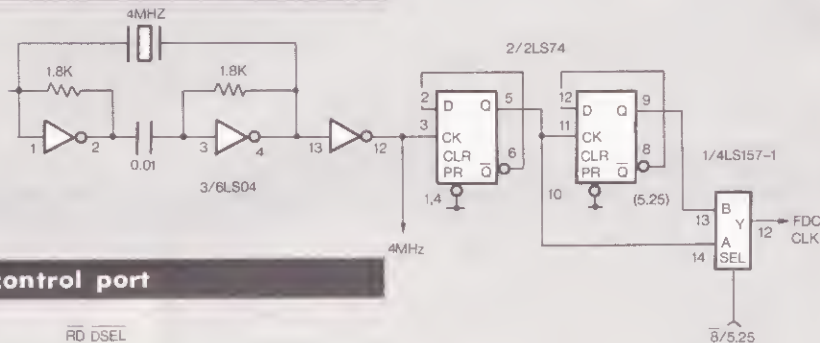
#### 回路

回路図を図1、(a)~(g)に示します。以下、この回路の大まかな説明をしましょう。(a)はアドレスデコーダで、I/O命令で28H~2BHが指定されたときFDCがアクティブに、2CHが指定されたときDSELがアクティブになります。(b)はクロックジェネレータです。FDCや5.25インチ用インタフェースが必要とするクロックを発生します。FDCが必要とするクロックは8インチと5.25インチでは異なるため、分周回路とデータセクタ(LS 157)を用いています。(c)はコントロールポートです。FDCからのIRQ、DRQのデータを2CHのリード時に出力するのが(c)の左側の回路で、2CHへのライトデータをラッチするのが右側の回

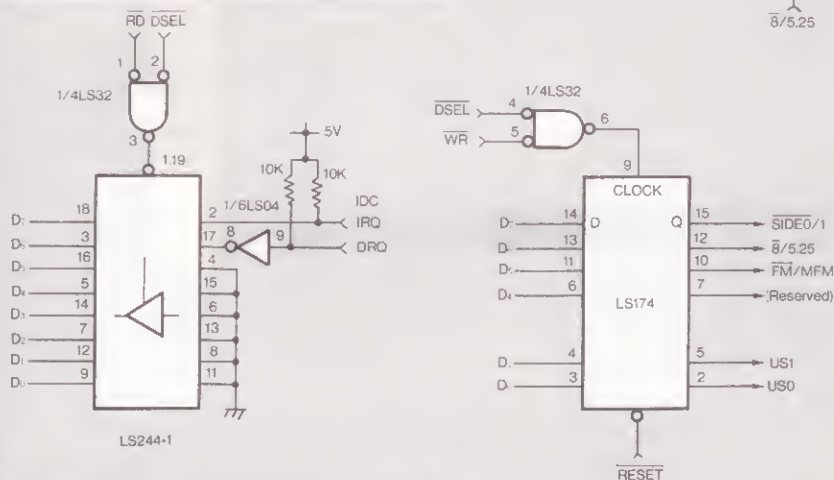
図1 (a)address decoder



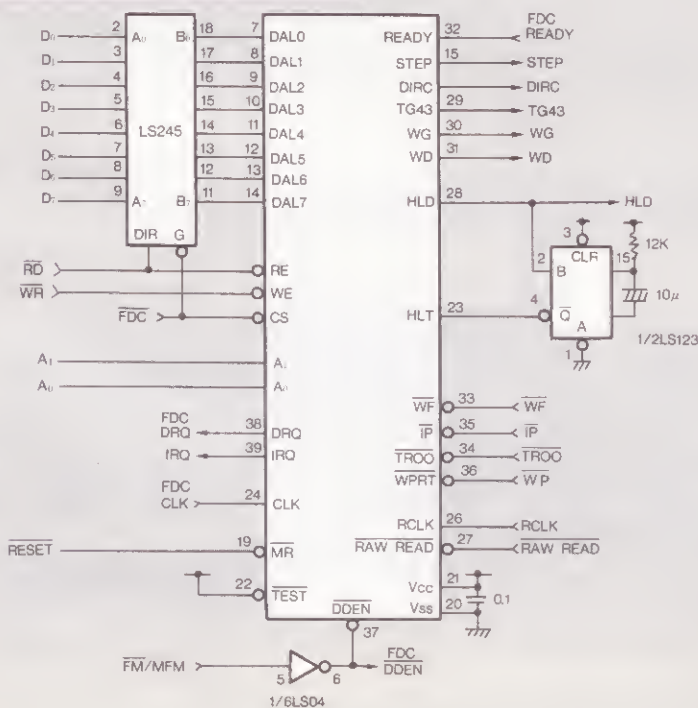
(b)clock generator



(c)control port



(d)FDC



路です。US0~US1はユニットセレクト信号で、8/5.25インチディスクについてそれぞれ4台までコントロールできます。つまり最大で8インチ4台、5.25インチ4台の計8台までコントロールできます。

(d)がFDC (floppy disk controller) で、8877というLSIを使用しました。LS123は、ヘッドをロード (ディスク面に押し付ける) してから安定するまでの50msのタイマーです。

(e)は8インチ用のインタフェースで、Y-E DATAのYD-180-1603というドライブを接続する回路です。

(f)は5.25インチ用にY-E DATAのYD-580を接続する回路です。

(g)は8インチ用の信号と、5.25インチ用の信号とを切り替えるためのセクタです。以上が大まかな回路の説明ですが、よく分からない人は参考文献の1を参照してください。

## 製作

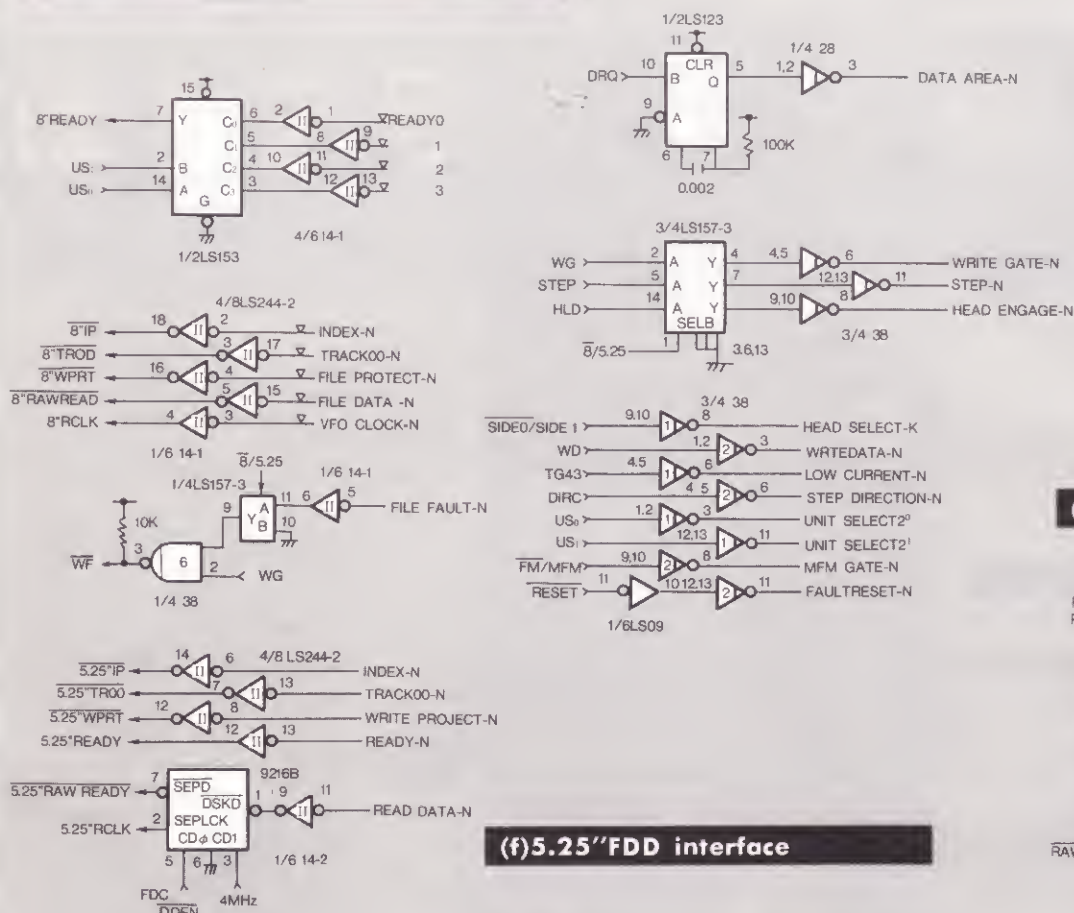
この回路のとおりに製作すればよいのですが、注意を少々。TTL数個に1個の割合で0.02~0.1μFくらいのセラミックコンデンサを+5VとGND(0V)の間に接続してください。これは、バイパスコンデンサと呼ばれ、TTLのスイッチング時のノイズを吸収するものです。これがない場合には、正しく回路を組んであっても誤動作する場合がありますので、注意してください。バイパスコンデンサは回路図には描かれていませんが、デジタル回路を製作するうえでの常識だと思ってください。

図2に私が作ったインタフェースボードの写真を示します。IC、LSIにはすべてソケットを使用していますが、本当に必要なのは8877と9216だけで、ほかのTTLは基板に直接ハンダ付けして結構です。

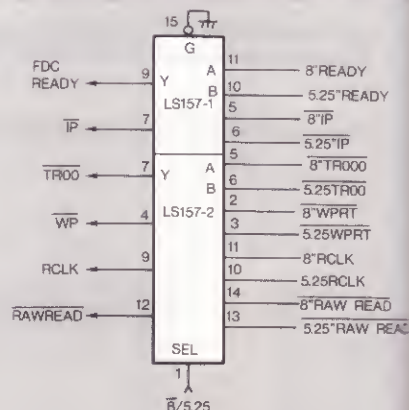
このインタフェースの電源はSMC本体から供給されるのでよいのですが、ドライブ自体には別の電源が必要です。図3に示しますのでドライブと一緒に購入してください。そのときに電源用のコネクタも買っておきましょう。また、インタフェースとドライブの接続にはフラットケーブルを使用します。8/5.25ではそれぞれ50/34芯を使用し、4台までのデジチェーン接続ができます。各ピン配置を図4に示します。



**(e) 8' FDD interface**



**(g) 8'-5.25" SELECTOR**



注)  $\nabla$ : 150 $\Omega$  pull up  
LS244-2 G(1, 19)はGNDに接続する  
LS157-1~3 SELおよびGは共通  
は7438あるいは7406使用

## 図2 インタフェースポート

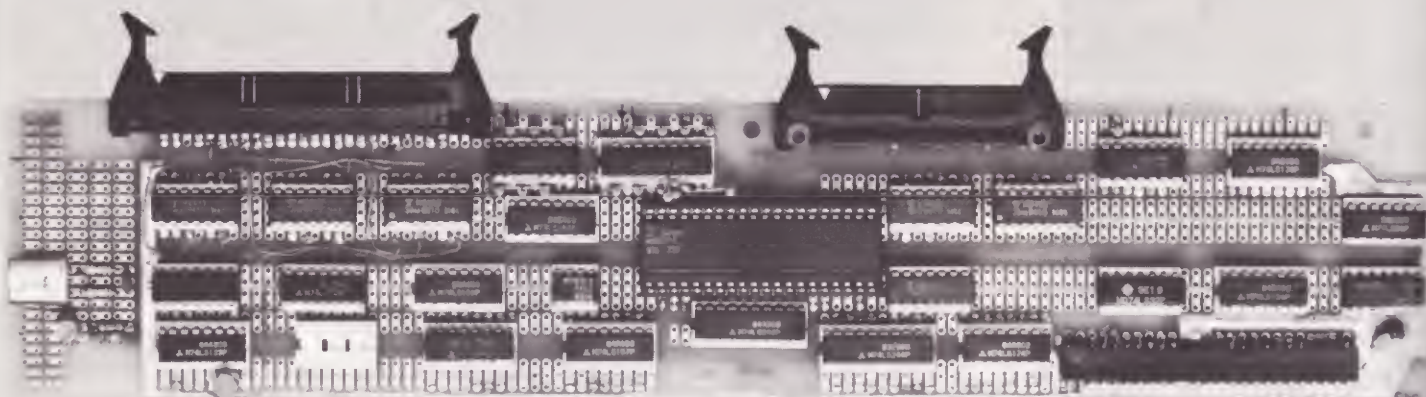


図3 YD-180-1603

電 源	電 源 容 量 (A)				
		1台	2台	3台	4台
+24V±10%	T Y P	0.5	1.0	1.5	2.0
	MAX	定常時	1.0	1.7	2.4
		起動時	1.0	2.0	3.0
+5V±5%	T Y P	1.2	2.2	3.2	4.2
	M A X	1.6	2.8	4.0	5.2

- (注1) MAX値は、電圧上限、周囲温度下限において、1台のみDRIVE SELECTされ、ライト中で他のドライブはDRIVE SELECT されていないときの値を示す。
- (注2) TYP 値は参考値で、電圧中心値、常温において、1台のみDRIVE SELECTされ、連続してランダムシーク、ライト、リードしているときの平均値を示す。ただし、他のドライブは、DRIVE SELECTされていないものとする。
- (注3) 電圧はドライブ受端で規定する。
- (注4) 起動時とはスピンドルモータが回転し始めた後の2秒間をいう。

図4 YD180-1603

信 号 名	ピン 番 号	信 号 名
GND	1	UNIT SELECT 2 <sup>1</sup> -N
UNIT SELECT2 <sup>0</sup> -N	3	GND
DISKETTE SENSE-N	5	GND
INDEX-N	7	GND
STEP-N	9	GND
LOW CURRENT-N	11	GND
WRITE GATE-N	13	GND
HEAD 1 SELECT-N	15	GND
WRITE DATA-N	17	GND
READY 0-N	19	GND
READY 1-N	21	GND
READY 2-N	23	GND
READY 3-N	25	GND
TRACK 00-N	27	GND
FILE PROTECT-N	29	GND
STEP DIRECTION-N	31	GND
VFO CLOCK-N	33	GND
SEPARATED DATE-N	35	GND
DATE AREA-N	37	GND
SEPARATED CLOCK-N	39	GND
MFM GATE-N	41	GND
	43	GND
	45	GND
FAULT-N	47	FAULT RESET-N
HEAD ENGAGE-N	49	KEY

SEPARATE DATA-Nは、FDD内部のジャンパ線を、SD から FD に変更して、FILE DATA-N として使用する

電源コネクタ (FDC側)  
日本AMP製P/N  
●ハウジング: 171822-7  
●コンタクト: 17020A-1  
●ケーブル: AWG20 3m以下

1	+24V
2	GND
3	+5V
4	GND
5	(OPEN)
6	GND
7	FG

FDD側	3M 3433
FDC側	3M 3425

## BIOS 変更

以上で製作したインタフェースをCP/M上で使用するにはソフトが必要ですから、BIOSを変更します。ただし、変更前のBIOSが本来もっていた機能(エスケープシーケンスなど)は可能なかぎり残すようにします。でなければ、スクリーンエディタなどスクリーンオリエンテッドなユーティリティプログラムが正常に働かなくなってしまう。また、いろいろなフォーマットのディスクに対応するようにしておけば、いろいろな機種のディスクをCP/Mで直接リード/ライトできるようになります。以上のことを考えて作ったのが、リスト1のプログラムです。エスケープシーケンスはほとんど変更はありませんが、ESC@のジェンロックモードは削除しました。また、Ctrl-Cを押したときなどのリブートによってファンクションキーの設定がクリアされてしまう機能を削除し、ESC 1を追加しました。せっかくESC 0を用いて定義したファンクションキーの内容がCtrl-Cを押しただけで消えてしまうのではどうしようもありませんから。この変更後のBIOSで使用できるエスケープシーケンスを図5に示します。

キー入力の割込モードは削除しました。割込キー入力で先行入力ができるのはディスク入出力を行っていないときだけで、ディスク入出力時はタイミングがぎりぎりです。ディスクは現在のところA、Bが3.5インチでCが8インチPC-9801、Dが5.25インチPC-9801としてあります。

将来的にはあらゆるフォーマットに対してリード/ライトできるようにするつもりです。また、RAM DISKも現在製作中で、完成したときには、Oh! HiTBiT誌上に発表したいと思います。乞御期待。

## 参考文献

- 1) 神崎康宏 「作りながら学ぶ マイコン設計トレーニング」 CQ出版社
- 2) 富士通 「MB 8866 ユーザ マニュアル」
- 3) Y-E DATA 「YD-180 両面フロッピーディスク装置」
- 4) ソニー 「SMC-777 ハードウェア解説書」

YD-580

ピン 番 号	コントローラ→FDD	YD-580
2		Reserved
4	→	Head Load
6	→	Drive Select 3
8	←	Index
10	→	Drive Select 0
12	→	Drive Select 1
14	→	Drive Select 2
16	→	Motor On
18	→	Direction Select
20	→	Step
22	→	Write Data
24	→	Write Gate
26	←	Track 00
28	←	Write Protect
30	←	Read Date
32	→	Side 1 Select
34	←	Ready



図5 エスケープシーケンス一覧表

制 御 内 容		エスケープコード	16進コード
カーソル制御	カーソルを1行分上へ移動	ESC A	1B 41H
	カーソルを1行分下へ移動	ESC B	1B 42H
	カーソルを1字分右へ移動	ESC C	1B 43H
	カーソルを1字分左へ移動	ESC D	1B 44H
	カーソルオン/オフ切り換え	ESE E v	1B 45H v
	カーソルを画面左上隅へ戻す	ESC H	1B 48H
	カーソルをr行c桁へ移動	ESC Y rc	1B 59H rc
	カーソルの位置を読み出す	ESC 5	1B 35H
文字数	40字/行モードにセット	ESC F	1B 46H
	80字/行モードにセット	ESC G	1B 47H
	文字数モードを読み出す	ESC ?	1B 3FH
キャラクタライン制	カーソル位置からその行の最後までをクリア	ESC I	1B 49H
	カーソル位置から画面右下隅までをクリア	ESC J	1B 4AH
	全キャラクタをクリア、左上隅にカーソル移動	ESC L	1B 4CH rc
	指定位置のキャラクタコードを読み出す	ESC <rc	
	カーソルの位置に1行追加	ESC N	1B 4EH
	カーソルのある行を削除	ESC O	1B 4FH
	カーソルの位置にスペースを挿入	ESC P	1B 50H
カラー制御	カーソルの位置のキャラクタを削除	ESC Q	1B 51H
	キャラクタのバックグラウンドカラーをセット	ESC 6 c	1B 36H c
	キャラクタカラーをセット	ESC 7 c	1B 37H c
	ボーダーエリアのカラーをセット	ESC 8 c	1B 38H c
アトリビュート制御	グラフィック表示面のカラーをセット	ESC ;c	1B 38H c
	文字反転/ブリンクの切り替え	ESC 3 a	1B 33H a
	現在のアトリビュートをセット	ESC 4 a	1B 34H a
その他	指定位置のアトリビュートを読み出す	ESC >rc	1B 3EH rc
	スクロールモードの切り替え	ESC =v	1B 3DH v
	画面上のキャラクタをプリンターに出力	ESC W	1B 57H
	記号キー、ファンクションキーに文字列をセット	ESC OKI	1B 30H kl
		SoSIS2...	
	キーボードバッファの内容をクリア	ESC 2	1B 32H
	サウンド発生	ESC 9dt	1B 39H dt
	Function key Reset	ESC !	1B 31H

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	Ascii	Dump
0400:	04	18	D2	CD	A3	F8	CD	97	F8	3A	83	EB	7D	20	EB	21	193		.....
0410:	26	F9	CD	E4	F8	CD	9E	0E	07	AB	E6	7E	C3	32	00	01	1E		.....
0420:	21	07	EA	22	01	00	32	05	0D	AB	E6	7E	C3	32	00	01	1E		.....
0430:	80	00	CD	AA	F4	CD	9D	CD	62	CD	62	CD	62	CD	62	CD	62		.....
0440:	04	00	4F	3A	4F	EB	7D	20	EB	D4	05	21	40	9B	4F	0C	97		.....
0450:	06	01	11	07	D4	EB	7D	20	EB	7D	20	EB	7D	20	EB	7D	20		.....
0460:	00	D4	21	33	EA	16	3E	10	CD	94	EC	7E	CD	94	EC	7E	CD		.....
0470:	27	1E	07	CD	96	EC	7E	CD	1A	23	1D	20	F6	D8	1E	43	30		.....
0480:	04	2B	FA	AF	CD	94	EC	15	20	CD	3A	6B	EB	07	00	3E	D6		.....
0490:	81	D3	1B	C9	D7	1B	D8	1B	E6	02	20	FA	C9	D7	1B	E6	02		.....
04A0:	11	72	EB	01	07	00	ED	80	CD	C9	EC	D1	F0	CD	C8	0B			.....
04B0:	F2	3A	77	EB	C6	20	4F	CD	C9	F0	3A	77	EB	C6	20	4F			.....
04C0:	78	EB	C6	20	4F	CD	C9	F0	3A	77	EB	C6	20	4F	CD	C9			.....
04D0:	62	F8	E6	80	F5	C4	FA	F0	F1	CC	B7	F1	C9	46	23	7E	D4		.....
04E0:	D3	1B	23	7E	D3	19	F6	23	7E	D3	19	F6	23	7E	D3	19	F6		.....
04F0:	23	46	23	7E	D3	19	F6	23	7E	D3	19	F6	23	7E	D3	19	F6		.....
Sum :	50	E1	EE	62	EF	EB	AF	D6	BE	0A	8A	93	CF	52	21	D6	:D7		

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	Ascii	Dump
0500:	04	18	D2	CD	A3	F8	CD	97	F8	3A	83	EB	7D	20	EB	21	193		
0510:	26	F9	CD	E4	F8	CD	97	F8	3A	83	EB	7D	20	EB	21	193			
0520:	21	07	EA	22	01	00	32	05	0D	AB	E6	7E	C3	32	00	01	1E		
0530:	80	00	CD	AA	F4	CD	9D	CD	62	CD	62	CD	62	CD	62	CD	62		
0540:	04	00	4F	3A	4F	EB	7D	20	EB	21	193	EB	7D	20	EB	21	193		
0550:	06	01	11	07	D4	EB	7D	20	EB	21	193	EB	7D	20	EB	21	193		
0560:	00	D4	21	33	EA	16	3E	10	CD	94	EC	7E	CD	94	EC	7E	CD		
0570:	27	1E	07	CD	96	EC	7E	CD	1A	23	1D	20	F6	D8	1E	43	30		
0580:	04	2B	FA	AF	CD	94	EC	15	20	CD	3A	6B	EB	07	00	3E	D6		
0590:	81	D3	1B	C9	D7	1B	D8	1B	E6	02	20	FA	C9	D7	1B	E6	02		
05A0:	11	72	EB	01	07	00	ED	80	CD	C9	EC	D1	F0	CD	C8	0B			
05B0:	F2	3A	77	EB	C6	20	4F	CD	C9	F0	3A	77	EB	C6	20	4F			
05C0:	78	EB	C6	20	4F	CD	C9	F0	3A	77	EB	C6	20	4F	CD	C9			
05D0:	62	F8	E6	80	F5	C4	FA	F0	F1	CC	B7	F1	C9	46	23	7E	D4		
05E0:	D3	1B	23	7E	D3	19	F6	23	7E	D3	19	F6	23	7E	D3	19	F6		
05F0:	23	46	23	7E	D3	19	F6	23	7E	D3	19	F6	23	7E	D3	19	F6		
Sum :	50	E1	EE	62	EF	EB	AF	D6	BE	0A	8A	93	CF	52	21	D6	:D7		

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump
0600:	FE	1B	28	51	FE	08	CA	D2	F0	FE	09	28	2A	FE	07	28	1A			
0610:	75	FE	0C	CA	95	F1	FE	0C	28	25	FE	07	28	2F	C9	21	45			
0620:	65	EB	11	33	F9	1A	3E	12	BE	38	1E	70	30	12	C3	45	D8			
0630:	F7	AF	72	F2	F9	1B	00	01	32	F7	ED	EB	F6	08	77	10				
0640:	CD	F2	F2	03	ED	F2	01	10	10	C3	F7	ED	AF	1B	02	3E	1B			
0650:	80	32	8C	EB	C9	21	A2	EF	18	44	21	EE	ED	1B	7F	21	69			
0660:	11	EF	18	4A	21	A5	EF	18	35	21	A4	EF	18	30	21	C5	16			
0670:	EF	18	4B	21	D2	EF	18	26	21	DD	EF	18	21	21	E5	EF	6D			
0680:	1B	1C	21	20	F0	1B	17	1B	10	12	1B	12	12	95	F0	1B	C8			
0690:	0D	21	49	00	1B	08	21	DC	F0	1B	03	21	92	F2	12	7F	05			
06A0:	EB	09	79	FE	20	7B	8C	FE	5B	20	AF	5B	20	87	21	EB	E7			
06B0:	EB	85	6F	30	01	3A	5E	23	56	EB	E9	5F	5E	5F	69	CD				
06C0:	EF	64	6E	69	EA	6E	6E	6E	6E	73	7E	7E	7E	7E	5A	19				
06D0:	EB	82	EB	87	EB	8C	EB	91	EE	AD	F0	5A	EE	D0	F0	C6	26			
06E0:	F0	CD	F0	D2	F0	5A	EE	F0	26	F1	5C	F1	5F	F1	8C	26				
06F0:	F1	5A	EE	95	F1	5A	EE	9D	F1	D0	F1	D0	F1	21	F2	5A	A1			
Sum :	94	75	7A	1E	15	54	BC	C7	1F	A1	C1	02	EB	AF	C5	D9	:44			

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	Ascii	Dump
0700:	04	18	D2	CD	A3	F8	CD	97	F8	3A	83	EB	7D	20	EB	21	193		
0710:	26	F9	CD	E4	F8	CD	97	F8	3A	83	EB	7D	20	EB	21	193			
0720:	21	07	EA	22	01	00	32	05	0D	AB	E6	7E	C3	32	00	01	1E		
0730:	80	00	CD	AA	F4	CD	9D	CD	62	CD	62	CD	62	CD	62	CD	62		
0740:	04	00	4F	3A	4F	EB	7D	20	EB	21	193	EB	7D	20	EB	21	193		
0750:	06	01	11	07	D4	EB	7D	20	EB	21	193	EB	7D	20	EB	21	193		
0760:	00	D4	21	33	EA	16	3E	10	CD	94	EC	7E	CD	94	EC	7E	CD		
0770:	27	1E	07	CD	96	EC	7E	CD	1A	23	1D	20	F6	D8	1E	43	30		
0780:	04	2B	FA	AF	CD	94	EC	15	20	CD	3A	6B	EB	07	00	3E	D6		
0790:	81	D3	1B	C9	D7	1B	D8	1B	E6	02	20	FA	C9	D7	1B	E6	02		
07A0:	11	72	EB	01	07	00	ED	80	CD	C9	EC	D1	F0	CD	C8	0B			
07B0:	F2	3A	77	EB	C6	20	4F	CD	C9	F0	3A	77	EB	C6	20	4F			
07C0:	78	EB	C6	20	4F	CD	C9	F0	3A	77	EB	C6	20	4F	CD	C9			
07D0:	62	F8	E6	80	F5	C4	FA	F0	F1	CC	B7	F1	C9	46	23	7E	D4		
07E0:	D3	1B	23	7E	D3	19	F6	23	7E	D3	19	F6	23	7E	D3	19	F6		
07F0:	23	46	23	7E	D3	19	F6	23	7E	D3	19	F6	23	7E	D3	19	F6		
Sum :	6A	4E	D1	01	6A	4E	D1	01	6A	4E	D1	01	6A	4E	D1	01	:67		

Adrs:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*			
0700:	EE	5A	EE	5A	EE	5A	EE	5A	EE	5A	EE	5A	EE	9B	EE	5A	:7E		Z	Z	Z	:S4Z	:Z	
0710:	EE	79	66	20	38	7C	EE	06	70	53	CD	5A	EE	02	22	79	EE	:28	y	B	80	:B8hady	:	
0720:	21	25	EF	18	19	79	EE	06	23	22	79	EE	28	26	FE	10	:30	EE	:	3	3	3	:B(3&0)	0
0730:	2A	79	EE	77	32	7B	EE	06	23	22	79	EE	21	41	EF	C3	:9E	FE	:	3	3	3	:y3A	3
0740:	EE	2A	79	EE	7B	7C	EE	05	71	23	22	79	EE	21	7B	EE	:7B	EE	:	3	3	3	:y3A	3
0750:	35	00	C3	5A	EE	21	6E	0A	11	0F	00	01	80	06	3A	01	:57		5	5	5	:y57	5	
0760:	23	71	00	19	10	7B	C3	5A	EE	CD	71	EF	CD	AF	B8	18	:81		3	3	3	:y3A	3	
0770:	E1	CD	B2	FE	21	00	00	22	48	EE	CD	40	EA	CD	02	EE	:E5		3	3	3	:y3A	3	
0780:	C8	CD	0E	EE	C7	79	21	75	EE	FE	44	20	02	CD	EE	:2E		3	3	3	:y3A	3		
0790:	45	00	02	CD	EE	FE	48	20	02	CD	EE	FE	49	20	02	CD	:3D		E	E	E	:yE	E	
07A0:	F6	C3	CD	F2	79	32	B1	EE	18	18	CD	71	EF	3A	33	59	:50		3	3	3	:y3A	3	
07B0:	C6	20	4F	CD	BA	ED	3A	32	79	60	20	4F	CD	BA	ED	:84		3	3	3	:y3A	3		
07C0:	AF	B8	C3	5A	EE	79	D6	20	67	07	07	E6	18	32	72	EE	:BF		3	3	3	:y3A	3	
07D0:	18	08	79	D6	20	E6	07	32	74	EB	C3	B8	79	D6	20	:FC		3	3	3	:y3A	3		
07E0:	D3	23	C3	5A	EE	79	32	7C	EB	21	EF	EE	C3	9E	EE	:79	:DA		3	3	3	:y3A	3	
07F0:	D6	20	7B	1F	5A	79	3C	EE	B6	20	18	17	49	29	60	:58		3	3	3	:y3A	3		
Sum :	B2	AC	F9	7F	41	00	B5	79	6D	EF	5F	9A	57	EC	71	:09	:EB							



[illegible]



# SMCリンクパッケージ集

西田 明宏

HiT BiTマイコンユーザーズクラブ

新井隆浩

リンクパッケージ集の第2回です。前回の記事に対して予想外の反響が得られ、驚いています。また投稿も多く集まり、選ぶのに苦労するほどです。さて、今月はFM系のBASICに装備されている大変便利なコマンドであるSYMBOLコマンド、簡単な時計機能を実現したTIMEプログラムの2点を紹介しましょう。

by HiT BiTマイコン  
ユーザーズクラブ

FMシリーズには、画面上の任意の位置に文字列を表示するSYMBOL命令があります。この命令は文字列をサイズや角度を変えて表示することができ、大変便利なものです。SMCのBASICには、グラフィック画面上に文字列を表示する機能はありません。そこでSYMBOLコマンドを作成しました。

## 人 力

リスト1にANN表記のソースリストを、リスト2にダンプリストを示します。リスト1を打ち込んでAssemでアセンブルするか、マシン語入力ツールまたはDebugでリスト2を打ち込んでください。

## 使い方

○書式

CALL SYMBOL (x, y, str, xm, ym, c, r, l)

x, y : 表示位置 (グラフィック座標)  
文字列の最初の文字の左上の座標を指定します。

str : 表示させる文字列

xm, ym : 文字列を拡大するときの倍率  
(1以上の整数)

c : カラーコード ( $0 \leq c \leq 15$ )

r : 回転角度 ( $0 \leq r \leq 3$ )  
時計回り 90度単位

1 : 論理演算コード (0 ≤ コード ≤ 4)  
BASIC のグラフィックス命令  
における論理演算コードと同じ。  
変数はすべて整数形です。また、座標と文  
字列以外は省略することができます。この  
コマンドではキャラクタの PCG に登録さ  
れているパターンをそのまま出力しますの  
で、DEF FONT 命令などで PCG を変え  
れば好きな図形を表示することができます。

## 使用例

例として、前号で紹介された HGETPUT  
コマンドと組み合わせて簡単なゲームを作  
ってみました。リスト 3 を入力してくださ  
い。これは戦車を操縦して地面に埋まっ  
ている地雷を避けながらベースまで行くゲ  
ームです。戦車の前後左右のいずれかに地雷  
が埋まっていると、画面右下に、"IMINE"  
と表示されます。斜めにある地雷は発見す  
ることはできません。所々に防衛電波を発  
している電波塔が置かれており、この近く  
に来ると地雷を捜すことができなくなり、  
モニターには ? MINE と表示されます。カ  
ーソルキーで自分の戦車を動かします。地  
雷に当たらずにベースへ行くことができれ  
ば、次の面に進みます。地雷にあたった場  
合はその面の地雷の位置が表示され、その  
面の最初に戻ります。また、69 回以内にベ  
ースまでたどりつけなかった場合はタイム  
オーバーとなり戦車が 1 台減ってしまいま  
す。このプログラムでは、標準ディスクに  
ついている EFFECT.PAC および前回の  
HGETPUT.PAC、今回の SYMBOL.PAC  
を使っていますので、走らせる前にこれら

をリンクしておいてください。  
SYMBOL コマンドは、FM 系の BASIC  
のプログラムを移植する際に一番困るコマ  
ンドでしたがもう大丈夫です。大変便利な  
コマンドですので、みなさん活用してくだ  
さい。

## TIME コマンド 新井隆浩

「SMC リンクパッケージ集」ができて、ゼ  
ひ何か投稿しようと思って、SMC-777(C)  
に欠けている時計機能を作ってみました。

## 人 力

リスト 4 にソースリストを、リスト 5 にダ  
ンプリストを示します。前の場合と同様に  
入力してください。ファイル名は TIME.  
PAC とします。

## 使い方

リンクパッケージをリンクすると、  
TCOUNTON, TCOUNTOFF というコマ  
ンドが使えるようになります。

書式

—TCOUNTON (文字形変数)

文字形変数にはあらかじめ数字 6 文字の文  
字列を代入しておきます。上から、HHM-  
MSS というようにそれぞれ時間、分、秒を  
表し、最初の値が初期値となります。たと  
えば 1 時 2 分 3 秒からスタートさせるには、  
T\$ = "010203"

—TCOUNTON (T\$)

とすればよいことになります。

その後、T\$ は時間に応じて書き直されま  
すから

PRINT T\$

でそのときの値が表示されます。また時間  
をセットしなおすときには

T\$ = "HHMMSS"

とします。

—TCOUNTOFF

で時計が止まります。

## 注意点

このプログラムでは割り込みを利用してい  
ますから SOUND 文および割り込みを使  
ったプログラムは時計を止めてから使っ  
てください (さもないと暴走する可能性があ  
ります)。時計を動かしているときには T  
\$ を参照するのはかまいませんが、T\$ に  
6 文字の数字の文字列以外を代入するこ  
とはやめてください。

なかなか正確な時間が得られますので、ゼ  
ひ使ってみてください。この時計は 24 時間  
方式です。リスト 6 に簡単なサンプルプロ  
グラムを示します。

僕の作ったプログラムが少しでもみなさん  
の役に立ってくれるといいなと思います。

当コーナーではひきつづき便利なリンクパ  
ッケージの募集を行っています。リンクパ  
ッケージに限らず、Sony-FILER や CP/  
M 上の便利なショートプログラムでもか  
まいません。採用分には本誌規定の原稿料  
をお支払いいたします。たくさんの投稿を  
お待ちしております。



```

10000 ;
10010 ; SYMBOL (<X>,<Y>,<STRING>,<dX>,<dY>,<Color>,<rotate>,<function>)
10020 ;
10030 ; program No.00017 written by Hisao.Nagaya.
10040 ;
10050 ; Copyright 27-Oct-1984
10060 ;
10070 ; ORG 0100H
10080 ; ENTRY SYMBOL (INT,INT,STR,[BYT],[BYT],[BYT],[BYT],[BYT])
10090 ; START
10100 ;
10110 ; BOXF EQU 0D85H
10120 ;
10130 ; CURX EQU 0FF44H
10140 ; CURY EQU 0FF46H
10150 ;
10160 ; SYMBOL: A=(HL)
10170 ; HL+
10180 ; H=(HL)
10190 ; L=A
10200 ; (CURX)=HL
10210 ; DE:HL
10220 ; E=(HL)
10230 ; HL+
10240 ; D=(HL)
10250 ; (CURY)=DE
10260 ; H=B
10270 ; L=C
10280 ; E=(HL)
10290 ; HL+
10300 ; D=(HL)
10310 ; HL+
10320 ; #=DE
10330 ; E=(HL)
10340 ; HL+
10350 ; D=(HL)
10360 ; HL+
10370 ; A=(IX+03H)
10380 ; A=40H
10390 ; A=0
10400 ; ->PARA4,~@
10410 ; A=(DE)
10420 ; A-
10430 ; PARA4: (DELTAX)=A
10440 ; A=A-A
10450 ; (DELTAX)=A
10460 ; E=(HL)
10470 ; HL+
10480 ; D=(HL)
10490 ; HL+
10500 ; A=(IX+04H)
10510 ; A=40H
10520 ; A=0
10530 ; ->PARA5,~@
10540 ; A=(DE)
10550 ; A-
10560 ; PARA5: (DELTAY)=A
10570 ; A=A-A
10580 ; (DELTAY)=A
10590 ; E=(HL)
10600 ; HL+
10610 ; D=(HL)
10620 ; HL+
10630 ; A=(IX+05H)
10640 ; A=40H
10650 ; ->PARA6,~@
10660 ; A=(DE)
10670 ; (COLOR)=A
10680 ; PARA6: E=(HL)
10690 ; HL+
10700 ; D=(HL)
10710 ; HL+
10720 ; A=(IX+06H)
10730 ; A=40H
10740 ; ->PARA7,~@
10750 ; A=(DE)
10760 ; (ROTATE)=A
10770 ; PARA7: E=(HL)
10780 ; HL+
10790 ; D=(HL)
10800 ; A=(IX+07H)
10810 ; A=40H
10820 ; ->PARA8,~@

```

# SYMBOL COMMAND

## LIST 2

Addr:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
0200:	02	79	EE	4A	32	3A	02	32	5C	02	E1	D1	05	46	23	E5	:66		γ/J21.2 年、5月F#		
0210:	2A	85	02	22	87	02	CB	20	CB	11	CB	20	CB	11	CB	20	:D5		*, "■", と, と, と, と		
0220:	CB	11	2A	7F	02	22	81	02	ED	78	04	D5	06	08	C5	17	:44		*, " ", Ox, ., ナ,		
0230:	F5	30	22	2A	87	02	22	46	FF	ED	48	89	02	B7	ED	4A	:12		00)*", "F, Ck, 千00		
0240:	EB	2A	81	02	22	44	FF	ED	48	83	02	B7	ED	4A	ED	4B	:E0		*, "Do K., 千00K		
0250:	8C	02	CB	8E	02	2A	81	02	ED	48	83	02	37	ED	4A	22	:E5		■, 4, *, 0K, 700		
0260:	81	02	F1	C1	10	CB	2A	87	02	ED	48	89	02	37	ED	4A	:F1		1, 4, 千, 0K, 700		
0270:	22	87	02	C1	78	E6	07	20	A9	2A	83	02	23	29	29	29	:E7		"■, 千, 0, 0, #))		
0280:	44	4D	2A	7F	ED	4A	22	7F	02	E1	D1	05	C2	F9	2F		:2F		DM, 千00, 千, ツホ		
0290:	01	C9	00	00	00	00	00	00	00	00	00	00	00	00	00	07	:D1		., 00, 00, 00, 00, 00		
02A0:	00	AF	ED	73	FE	FF	31	FE	FF	D3	1D	CD	85	0D	ED	7B	:F1		., 00, 00, 00, 00, 00		
02B0:	FE	FF	3E	10	D3	1D	C9	00	00	A0	00	10	10	00	00	08	:CC		00), 00, 00, 00, 00		
02C0:	80	00	22	00	04	00	08	00	20	00	84	80	02	01	24	24	:1D		., 00, 00, 00, 00, 00		
02D0:	84	42	48	41	24	12	01	20	00	48	00	20	40	81	02	48	:19		■BHA*, .H, 0., H		
02E0:	82	04	41	00	80	41	04	00	00	00	00	1A	1A	1A	1A	1A	:F4		., A., A., 0., 0.		
02F0:	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	:A0		., 0., 0., 0., 0., 0.		
Sum :	E9	18	97	84	B3	BC	2F	B2	51	B1	0B	EB	AD	75	F6	69	:B5				

```

10830 A=(DE)
10840 (COLOR1)=A
10850 PARAB: HL=#
10860 B=(HL)
10870 HL+
10880 E=(HL)
10890 HL+
10900 D=(HL)
10910 DE:HL
10920 ;
10930 C=00000010B
10940 B=BC
10950 B=HL
10960 A=(ROTATE)
10970 A=A&00000011B
10980 HL=CURX0
10990 DE=CURY0
11000 BC=0000H
11010 A=2
11020 ->PLUSX,@
11030 B=B
11040 PLUSX: A=A&00000011B
11050 =>PLUSX,E
11060 C=B
11070 PLUSY: >{.}@
11080 ->NOEXX,~@
11090 DE:HL
11100 ;
11110 NOEXX: (LOOPY1)=HL ; CURX0
11120 (NOPL0T46)=HL
11130 (NOPL0T52)=HL
11140 HL+
11150 HL+
11160 (LOOPY4)=HL ; CURX1
11170 (NOPL0T1)=HL
11180 (NOPL0T11)=HL
11190 HL+
11200 HL+
11210 (LOOPX27)=HL ; DELTAX
11220 DE:HL
11230 (LOOPC5)=HL ; CURY0
11240 HL+
11250 HL+
11260 (LOOPC8)=HL ; CURY1
11270 (NOPL0T18)=HL
11280 (NOPL0T28)=HL
11290 HL+
11300 HL+
11310 (LOOPX13)=HL ; DELTAX
11320 A=B
11330 A=A^4AH
11340 (LOOPX17)=A
11350 (NOPL019)=A
11360 (NOPL0150)=A
11370 A=C
11380 A=A^4AH
11390 (LOOPX31)=A
11400 (NOPL0T26)=A
11410 HL=#
11420 BC=#
11430 ;
11440 LOOPC: B=BC
11450 B=(HL)
11460 HL+
11470 B=HL
11480 DB 2AH
11490 LOOPC5: DW CURY0 ; +5
11500 DB 22H
11510 LOOPC8: DW CURY1 ; +8
11520 <B0
11530 <@C
11540 <B0
11550 <@C
11560 <B0
11570 <@C
11580 LOOPY: DB 2AH
11590 LOOPY1: DW CURX0 ; +1
11600 DB 22H
11610 LOOPY4: DW CURX1 ; +4
11620 A=P(C)
11630 B+
11640 B=BC
11650 B=B
11660 LOOPX: B=BC
11670 <@.
11680 B=AF
11690 ->NOPL0T,~@
11700 DB 2AH
11710 LOOPX6: DW CURY1 ; +6
11720 DB 22H

```

```

11730 LOOPX9: DW CURY ; +9
11740 DW 4BEDH
11750 LOOPX13: DW DELTAY ; +13
11760 A=A^A
11770 DB 0EDH
11780 LOOPX17: DB 4AH ; +17
11790 DE:HL
11800 DB 2AH
11810 LOOPX20: DW CURX1 ; +20
11820 DB 22H
11830 LOOPX23: DW CURX ; +23
11840 DW 4BEDH
11850 LOOPX27: DW DELTAX ; +27
11860 A=A^A
11870 DB 0EDH
11880 LOOPX31: DB 4AH ; +31
11890 BC=(COLOR)
11900 ;
11910 JBOXF
11920 ;
11930 NOPL0T: DB 2AH
11940 NOPL0T1: DW CURX1 ; +1
11950 BC=(DELTAX)
11960 @=1
11970 DB 0EDH
11980 NOPL0T9: DB 4AH ; +9
11990 DB 22H
12000 NOPL0T11: DW CURX1 ; +11
12010 AF=#
12020 BC=#
12030 ~>LOOPX
12040 DB 2AH
12050 NOPL0T18: DW CURY1 ; +18
12060 BC=(DELTAY)
12070 @=1
12080 DB 0EDH
12090 NOPL0T26: DB 4AH ; +26
12100 DB 22H
12110 NOPL0T28: DW CURY1 ; +28
12120 BC=#
12130 A=B
12140 A=A&7
12150 ->LOOPY,~0
12160 ;
12170 HL=(DELTAX)
12180 HL+
12190 HL=HL+HL
12200 HL=HL+HL
12210 HL=HL+HL ; 8
12220 B=H
12230 C=L
12240 DB 2AH
12250 NOPL0T46: DW CURY0 ; +46
12260 A=A^A
12270 DB 0EDH
12280 NOPL0T50: DB 4AH ; +50
12290 DB 22H
12300 NOPL0T52: DW CURX0 ; +52
12310 ;
12320 HL=#
12330 BC=#
12340 B=
12350 ->LOOPC,~0
12360 I
12370 ;
12380 CURX0: DW 0
12390 CURX1: DW 0
12400 DELTAX: DB 0
12410 DELTAX1: DB 0
12420 CURY0: DW 0
12430 CURY1: DW 0
12440 DELTAY: DB 0
12450 DELTAY1: DB 0
12460 ROTATE: DB 0
12470 COLOR: DB 7
12480 COLOR1: DB 0
12490 ;
12500 UBOXF: A=A^A
12510 (OFFFEH)=SP
12520 SP=OFFFEH
12530 P(1DH)=A
12540 IBOXF
12550 SP=(OFFFEH)
12560 A=10H
12570 P(1DH)=A
12580 I
12590 ;
12600 END

```



# SYMBOL COMMAND

```

10000 ' Save "A:THE-MINE" ' LINK "EFFECT" & "SGPUT" & "SYMBOL"
10010 ' ===== THE MINE GAME '85.4.1 by MUC (K.Tanaka) =====
10020 DEF INT A=2: DIM PAT(8,64),CLS(8),YX(13,13),OT(4),PL(7,2)
10030 DEF FN X=(X-1)*16: DEF FN Y=(Y-1)*16: RANDOMIZE RND(-1)
10040 CURSOR OFF : WIPE: CONSOLE 40: GMODE 1: GOSUB *INIT: SC=0: DN=3: JR=10: TK=5: AR=1
10050 *MAIN : GOSUB *SET
10060 *LOOP : GOSUB *OTO
10070 B=INP(%HIB): IF B=8 THEN 10070 ELSE
10080 P=INSTR("H /",INKEY$(1)): IF P=0 THEN *LOOP
10090 IF P=1 AND Y=2 AND X=1 THEN *WIN
10100 ON P GOSUB *UP,*RIGHT,*DOWN,*LEFT
10110 IF YX(Y,X)=7 THEN GOSUB *BOM: GOTO *TEND ELSE IF SS>69 THEN *TEND
10120 GOSUB *MSG
10130 GOTO *LOOP
10140 *WIN : FOR I=0 TO 7: _SPUT(FNX,FNY,16,16-I*2,PAT(1,I*4)): _PANIC(1)
10150 _SPUT(FNX,FNY,(I+1)*2+16,16,2,PAT(0,0)): NEXT : SC=SC+10: SS=SS+1: _PANIC
10160 GOSUB *SCO: JR=JR+1: AR=AR+1: GOSUB *JRP: IF AR MOD 2 THEN DN=DN+1: TK=TK+1
10170 GOTO *MAIN
10180 *SCO : SC=SC+1: SS=SS+1: BOXF(265,96)-(295,102),0: IF SC>9999 THEN SC=9999
10190 _SYMBOL(264,96,SPACE$(4-LEN(STR$(SC)))+STR$(SC),1,1,7): RETURN
10200 *UP : IF Y=1 OR YX(Y-1,X)=5 OR YX(Y-1,X)=6 THEN RETURN
10210 FOR I=1 TO 8: _SPUT(FNX,FNY-I*2+16,16,2,PAT(0,0))
10220 _SPUT(FNX,FNY-I*2,16,16,PAT(1,0)): _PANIC(1): NEXT : Y=Y-1: GOSUB *SCO
10230 RETURN
10240 *DOWN : IF Y=12 OR YX(Y+1,X)=5 OR YX(Y+1,X)=6 THEN RETURN
10250 FOR I=1 TO 8: _SPUT(FNX,FNY+I*2-2,16,2,PAT(0,0))
10260 _SPUT(FNX,FNY+I*2,16,16,PAT(3,0)): _PANIC(1): NEXT : Y=Y+1: GOSUB *SCO
10270 RETURN
10280 *LEFT : IF X=1 OR YX(Y,X-1)=5 OR YX(Y,X-1)=6 THEN RETURN
10290 FOR I=1 TO 8: _SPUT(FNX-I*2+16,FNY,2,16,CLS(0))
10300 _SPUT(FNX-I*2,FNY,16,16,PAT(4,0)): _PANIC(1): NEXT : X=X-1: GOSUB *SCO
10310 RETURN
10320 *RIGHT : IF X=12 OR YX(Y,X+1)=5 OR YX(Y,X+1)=6 THEN RETURN
10330 FOR I=1 TO 8: _SPUT(FNX+I*2-2,FNY,2,16,CLS(0))
10340 _SPUT(FNX+I*2,FNY,16,16,PAT(2,0)): _PANIC(1): NEXT : X=X+1: GOSUB *SCO
10350 RETURN
10360 *BOM : _SPUT(FNX,FNY,16,16,PAT(8,0),0): _GUN
10370 MS1$="YOU": MS2$="DIE!": GOSUB *MSG2: RETURN
10380 *OTO : FOR I=0 TO 4: OUT 83,OT(I): NEXT : RETURN
10390 *MSG : IF YX(Y+1,X)=6 OR YX(Y-1,X)=6 OR YX(Y,X-1)=6 OR YX(Y,X+1)=6
      THEN MS1$="?": GOSUB *MSG1: RETURN
10400 IF YX(Y+1,X+1)=6 OR YX(Y-1,X-1)=6 OR YX(Y+1,X-1)=6 OR YX(Y-1,X+1)=6
      THEN MS1$="?": GOSUB *MSG1: RETURN
10410 IF YX(Y-1,X)=5 THEN MS1$="INTO": MS2$="BASE": GOSUB *MSG2: RETURN
10420 MS1$=STR$(YX(Y,X)): GOSUB *MSG1: RETURN
10430 *MSG1 : BOXF(213,118)-(230,161),0: _SYMBOL(211,130,MS1$,2,3,4): RETURN
10440 *MSG2 : BOXF(213,118)-(307,161),0: _SYMBOL(259-LEN(MS1$)*8,118,MS1$,2,3,4)
10450 _SYMBOL(259-LEN(MS2$)*8,141,MS2$,2,3,4): RETURN
10460 *TANK : J=0: IF TK>6 THEN J=1
10470 FOR I=1 TO TK-J: _SPUT(I*16+196,170,16,16,PAT(1,0),5): NEXT : _CAR(10)
10480 FOR I=6 TO TK-1 STEP -1: BOXF(I*16+212,170)-(I*16+235,185),9: NEXT
10490 RETURN
10500 *JRP : FOR Y=1 TO 12: FOR X=1 TO 12
10510 IF YX(Y,X)=7 THEN _SPUT(FNX,FNY,16,16,PAT(7,0),0): _UP3(10)
10520 NEXT : NEXT : FAUSE 10: RETURN
10530 *TEND : IF SS>69 THEN SC=SC-70: MS1$="COUNT": MS2$="OVER!": GOSUB *MSG2
10540 _DOWN3(255): TK=TK-1: GOSUB *JRP: IF TK>0 THEN *MAIN
10550 BOXF(213,118)-(307,161),0: _SYMBOL(234,123,"REPLAY",1,2,6)
10560 _SYMBOL(213,143,"GAME (Y/N) ?",1,2,6): EVAL INKEY$
10570 I=INSTR("YyNn",INKEY$(1))
10580 IF I THEN IF I<3 THEN SC=0: AR=1: DN=3: JR=10: TK=5: GOTO *MAIN ELSE
      IF I>2 THEN WIPE: FONT RESET: END
10590 GOTO 10570
10600 *SET : SC=SC-1: SS=-1: BOXF(273,72)-(287,78),0: IF TK>7 THEN TK=7
10610 _SYMBOL(272,72,SPACE$(2-LEN(STR$(AR)))+STR$(AR),1,1,7)
10620 _UP1(5): GOSUB *SCO: _UP1(5): MS1$="WAIT!": MS2$=" ": IF DN>7 THEN DN=7
10630 FOR Y=1 TO 12: FOR X=1 TO 12: YX(Y,X)=0: NEXT : NEXT : GOSUB *MSG2
10640 FOR A=123 TO 194: DEF FONT A,"FFFFFFFFFFFFFFFF": NEXT
10650 I=IRND(7):
10660 FOR Y=1 TO 12: FOR X=1 TO 12: _SPUT(FNX,FNY,16,16,PAT(0,0)): NEXT : NEXT
10670 _SGET(0,0,2,16,CLS(0)): YX(1,1)=5: _SPUT(0,0,16,16,PAT(5,0))
10680 FOR I=1 TO DN
10690 X=IRND(9)+2: Y=IRND(9)+2: IF YX(Y,X)=6 THEN 10690
10700 YX(Y,X)=6: _SPUT(FNX,FNY,16,16,PAT(6,0)): NEXT : IF JR>20 THEN JR=20
10710 FOR I=1 TO JR
10720 X=IRND(11)+1: Y=IRND(11)+1: IF YX(Y,X)<>0 OR X+Y>22 OR X+Y<5 THEN 10720
10730 YX(Y,X)=7: NEXT
10740 FOR Y=1 TO 12: FOR X=1 TO 12: I=0
10750 IF YX(Y,X)<>0 THEN 10810
10760 IF YX(Y-1,X)=7 THEN I=I+1
10770 IF YX(Y+1,X)=7 THEN I=I+1
10780 IF YX(Y,X-1)=7 THEN I=I+1
10790 IF YX(Y,X+1)=7 THEN I=I+1

```





```

11700 DATA 63,3C66461818001800, 64,1C36636B6E301E00, 65,1C3663637F636300
11710 DATA 66,7E63637E63637E00, 67,1E33606060331E00, 68,7C66636363637C00
11720 DATA 69,3F30303E30303F00, 70,7F60607E60606000, 71,1F30606763331F00
11730 DATA 72,6363637F63636300, 73,3F0C0C0C0C0C3F00, 74,0303030303633E00
11740 DATA 75,63666C787C6E6700, 76,3030303030303F00, 77,63777F7F6B636300
11750 DATA 78,63737B7F6F676300, 79,3E63636363633E00, 80,7E6363637E606000
11760 DATA 81,3E6363636F663D00, 82,7E6363677C6E6700, 83,3C66603E03633E00
11770 DATA 84,3F0C0C0C0C0C0C00, 85,6363636363633E00, 86,636363773E1C0800
11780 DATA 87,63636B7F7F362200, 88,63773E1C3E776300, 89,3333121E0C0C0C00
11790 DATA 90,7F070E1C38707F00, 91,7C60606060607C00, 92,66663C7E187E1800
11800 DATA 93,3E06060606063E00, 94,183C660000000000, 95,00000000000007E00
11810 DATA 96,3030180C00000000, 97,00003C063E663B00, 98,60607C6666663C00
11820 DATA 99,00003C6260623C00, 100,06063E6666663E00, 101,00003C667E603C00
11830 DATA 102,0C1A187E18181800, 103,00003E66663E063C, 104,60607C6666666600
11840 DATA 105,1800381818183C00, 106,060006060666663C, 107,6060666C787C6600
11850 DATA 108,38181818183C00, 109,0000766B6B6B6300, 110,00005F6363636300
11860 DATA 111,00003E6363633E00, 112,00007C66667C6060, 113,00003E66663E0606
11870 DATA 114,00006C7260606000, 115,00003E603C067C00, 116,18187E1818180E00
11880 DATA 117,0000666666663D00, 118,00006666663C1800, 119,0000636B6B6B3600
11890 DATA 120,0000663C183C6600, 121,00006666663E063C, 122,00007E0C18307E00,-1

```

#### LIST.4

```

10 ;-----
20 ;--- TIME Program Version 2.1 -----
30 ;----- Programed 1985/3/22 -----
40 ;----- by Takahiro arai -----
50 ;----- (SMC-777 Version) -----
60 ;-----
70 ; ORG 100H
80 ; ENTRY TCOUNT(STR)
90 ; ENTRY TCOUNTOFF
100 ;
110 ; START
120 ;
130 TCOUNT:
140 ; DI
150 ; HL+
160 ; (STRGADR)=HL
170 ; A=0
180 ; (TIMEC)=A
190 ; HL=(0FC01H)
200 ; (DATA)=HL
210 ; HL=TCOUNT
220 ; (0FC01H)=HL
230 ; IM1
240 ; A=1
250 ; F(21H)=A
260 ; EI
270 ; [
280 ;
290 TCOUNT:
300 ; #=HL
310 ; #=BC
320 ; #=AF
330 ; HL=TIMEC
340 ; A=(HL)
350 ; A+
360 ; A-60
370 ; ->CEND,~0
380 ; A=0
390 ; (HL)=A
400 ; HL=(STRGADR)
410 ; C=(HL)
420 ; HL+
430 ; H=(HL)
440 ; L=C
450 ; BC=0005
460 ; HL=HL+BC
470 ; A=(HL)
480 ; A+
490 ; A-3AH
500 ; ->CEND,~0
510 ; A=30H
520 ; (HL)=A
530 ; HL-
540 ; A=(HL)
550 ; A+
560 ; A-36H
570 ; ->CEND,~0
580 ; A=30H
590 ; (HL)=A
600 ; HL-
610 ; A=(HL)
620 ; A+
630 ; A-3AH
640 ; ->CEND,~0
650 ; A=30H
660 ; (HL)=A
670 ; HL-
680 ; A=(HL)
690 ; A+
700 ; A-36H
710 ; ->CEND,~0
720 ; A=30H
730 ; (HL)=A
740 ; HL-
750 ; A=(HL)
760 ; A+
770 ; A-34H
780 ; ->HEND,0
790 ; A-3AH
800 ; ->CEND,~0
810 ; A=30H
820 ; (HL)=A
830 ; HL-
840 ; A=(HL)
850 ; A+
860 ; ->CEND
870 ; HEND: (HL)=A
880 ; HL-
890 ; A=(HL)
900 ; A-32H
910 ; ->CEND,~0
920 ; A=30H
930 ; HL+
940 ; (HL)=A
950 ; HL-
960 ; CEND: (HL)=A
970 ; AF=#
980 ; BC=#
990 ; HL=#
1010 ; [
1020 ;
1030 TCOUNTOFF:
1040 ; DI
1050 ; A=0
1060 ; F(21H)=A
1070 ; HL=(DATA)
1080 ; (0FC01H)=HL
1090 ; [
1100 ;
1110 ; DATA: DS 2
1120 ; TIMEC: DS 1
1130 ; STRGADR: DS 2
1140 ;
1150 ; END

```

# TIME COMMAND

## LIST.5

Addr:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump	*
0100:	B3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:B3		.....	.....	
0110:	00	00	00	52	31	8B	00	24	00	00	01	54	43	4F	55	4E	:EC		...R17.5	...TCOUN	
0120:	54	4F	4E	00	27	01	24	01	54	43	4F	55	4E	54	4F	46	:B0		TON. 1.5	TCOUNTOF	
0130:	46	00	AA	01	26	01	00	08	00	00	F3	23	22	B9	01	3E	:50		F.1.&...	..月#7. >	
0140:	00	32	B8	01	2A	01	FC	22	B6	01	21	45	01	22	01	FC	:71		.27. *. 利"	カ. !E. " ; 利	
0150:	ED	56	3E	01	D3	21	FB	D9	E5	C5	F5	21	B8	01	7E	3C	:6D		CV>.モ! 町ノ	▲7 町! 7. ^<	
0160:	FE	3C	20	54	3E	00	77	2A	B9	01	4E	23	66	69	01	05	:8D		機< T>.w*	7.N#fi..	
0170:	00	09	7E	3C	FE	3A	20	40	3E	30	77	2B	7E	3C	FE	36	:59		.. ^< 機: @	>Ow+ ^< 機	
0180:	20	36	3E	30	77	2B	7E	3C	FE	3A	20	2C	3E	30	77	2B	:B4		6>Ow+ ^< 機:	,>Ow+	
0190:	7E	3C	FE	36	20	22	3E	30	77	2B	7E	3C	FE	34	28	0C	:60		^< 機6 "	>O w+ ^< 機4 (.	
01A0:	FE	3A	20	14	3E	30	77	2B	7E	3C	18	0C	77	2B	7E	FE	:7B		機: .>Ow+	^< ..w+ ^< 機	
01B0:	32	20	05	3E	30	23	77	2B	77	F1	C1	E1	C9	F3	3E	00	:8E		2 .>O#w+	wP# 1/月>.	
01C0:	D3	21	2A	B6	01	22	01	FC	C9	00	00	00	00	00	00	00	:BD		モ! *カ. ". 利	/.....	
01D0:	28	00	A0	10	82	40	00	20	02	00	00	00	00	00	00	00	:BC		(. . . @.	.....	
01E0:	00	00	00	00	40	00	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	:44		.....@.....	.....	
01F0:	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	:A0		.....	.....	
Sum :	EB	23	D1	7D	99	35	91	94	4F	00	C9	09	00	DA	B2	AE	:AA				

SMC

LINK

PACKAGES

## LIST.6

```

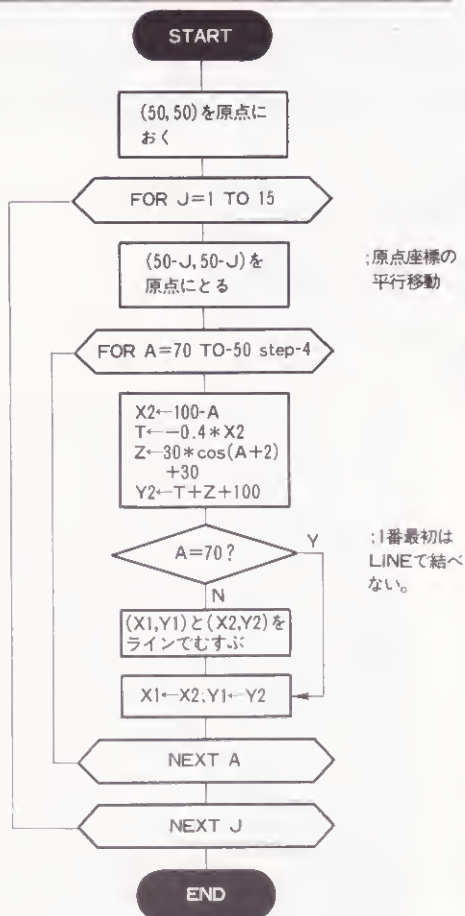
2  -- TIME コマンド Test Programme --
10 WIPE:CURSOR OFF :CONSOLE 40
20 T$="000000"
30 _TCOUNTON(T$)
40 LOCATE (10,10)
42 PRINT LEFT$(T$,2);CHR$(245);
44 PRINT MID$(T$,3,2);CHR$(246);
46 PRINT RIGHT$(T$,2);CHR$(247);
50 IF INP(&H1B) AND 4 THEN 6: ELSE GOTO 40
60 _TCOUNTOFF
    
```



# 見栄3次元CG講座(4)

いやー、そーいうわけで、ワタシのこの講座もついに最終回をむかえることになりました。ラストは有終の美を飾って、パツとハデに28ページほどの特集に、／——という声も特になく、ひっそりと消えていくばかりです。いえ、そーではありません。ま、いわば発展的解消というヤツでしょう。見栄3次元は、とりあえずこれで十分、細工は流流。今度は、本物の3Dグラフィックスに挑戦してほしいと思っておるワケなのです。そこで、今回は最後のサービス、カラーパレットの集大成をお届けすることにしました。しっかりね。それでは、またどこかでお会いすることにして——<sup>ツライチン</sup>再見!!

図2 リスト1のフローチャート



リスト1

```

10 WIPE
20 GLOCATE (50,50)
30 FOR J=1 TO 15
40 GLOCATE (50-J,50-J)
50 FOR A=70 TO -50 STEP -4
60 X2=100-A
70 T=-.4*X2
80 Z=30*COS((A+2)*PI/30)+30
90 Y2=T+Z+100
100 IF A=70 THEN 120
110 LINE (X1,Y1)-(X2,Y2),1
120 X1=X2:Y1=Y2
130 NEXT A
140 NEXT J
150 END
  
```



まず、GLOCATE 命令(グラフィックス表示面上の座標を、いろいろ動かしてしまう機能)から始めることにしましょう。

とりあえず、リスト1を打ち込んで実行してみてください。テープを折り曲げたような形のものが出てきましたね。

リスト1の行番号20は、実はあってもなくてもいいのですが……が、初めの原点が(50, 50)の位置にあることを示すために存在しています。

行番号40では、原点を1ずつ右斜め前方にずらしているのが分かりますね。

行番号60~90は、読者が自分でいろいろ変えてみてください。

つまり、ここで重要なのは、原点を少しずつ変化させて、同じことを数回繰り返しているだけ——という点なのです。また、GLOCATE 命令では、座標の符号を気にしなくてもいいというのも利点です。

次は、リスト2を打ち込んで実行してみてください。いかがですか？

あまり3次元に見えないかもしれませんね。一応、くぼみらしきものを作ってみたつもりなんですけどねー。

行番号20のGLOCATE 命令で、( )の次にふたつのパラメータ、-1と1があります。これはいったいナンなのかというと、X方向では画面の左側を正の方向、右側を負の方向とすることを意味し、Y方向について

# 図1 GLOCATEの説明

GLOCATE(X,Y),1,1

- ① 座標 X,Y  $-16384 \leq X,Y \leq 16383$   
初期座標系による値を示す
- ② 正の時 右方向を正の方向  
負の時 逆となる
- ③ 正の時 上方向を正  
負の時 逆となる

③で0以上の時

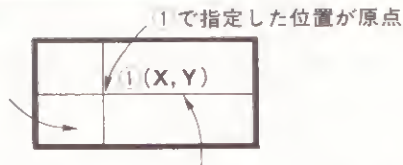
(負)

(正)

③で0未満の時

(正)

(負)



②で0以上の時 (負) → (正) ×

0 1 未満の時 (正) ← (負)

は、下方向を正、上方向を負とすることを意味しているのです(図1参照)。リスト2のテクニックは、行列の回転および直線の傾きを、三角関数のひとつTANを用いている点です。

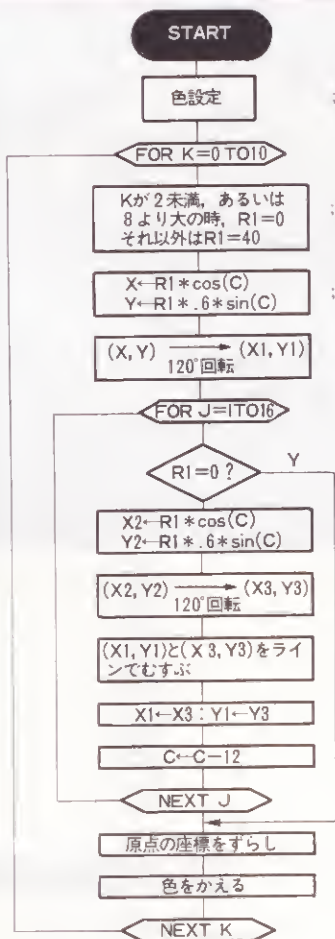
行番号130からも分かるように、傾き角は30度、行番号70のQ=120、C=45、D=12をうまく使って、連続した図形を得ています。リスト3に移りましょう。

今度は凸を表現してみたいと思います。リスト3を打ちこんだら、実行してみてください。いかがでしょう…といっても、カラーパレットの機能を用いているのは「暗→明→暗」だけです。

ここでのテクニックは、行番号90~120です。K=0~2及びK=13,14のときは、N=5でR1=0となり、凸はできずに直線を形成します。行番号100のN=N-1で、Nが小さくなるにつれてR1がだんだん大きくなり、N=0のとき最大値30をとります。それから後は、N=-1~-4となり、R1はだんだん小さくなっていきます。原理は、これだけです。

CやDの値については、C=135でcos(c)とsin(c)の関係が、cos(c)=-sin(c)であることを利用しました。D=12というのは、最も左右対称の図形ができるときの値が、12であったからにすぎません。ために他の値でやってごらん下さい。

## 図3 リスト2のフローチャート



緑の15階調

R1は基準の半径の大きさ

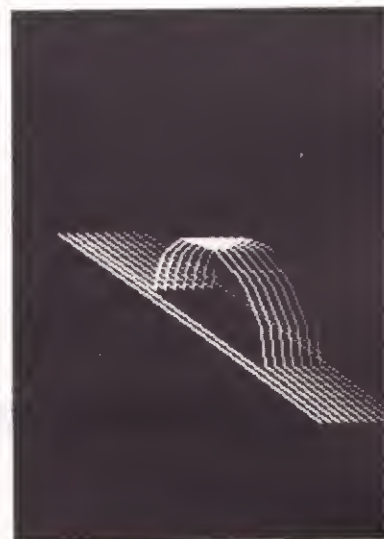
C=45(初期設定時)

## リスト2

```

10 WIDE:GMODE 1
20 GLOCATE (70,70),-1,1
30 FOR I=1 TO 15
40 DEF PALETTE I,0,I,0
50 NEXT I
60 G=5
70 D=12:Q=120
80 FOR K=0 TO 10
90 IF K<2 OR K=13 THEN R1=0 GOTO 110
100 R1=40
110 R2=R1*.6
120 S1=-70
130 T1=-S1*TAN(30*PI/180)
140 C=45
150 X=R1*COS(C*PI/180)
160 Y=R2*SIN(C*PI/180)
170 X1=X*COS(Q*PI/180)-Y*SIN(Q*PI/180)
180 Y1=X*SIN(Q*PI/180)+Y*COS(Q*PI/180)
190 S2=X1:T2=Y1
200 LINE (S1,T1)-(S2,T2),G
210 FOR J=1 TO 16
220 IF R1=0 THEN GOTO 240
230 X2=R1*COS(C*PI/180)
240 Y2=R2*SIN(C*PI/180)
250 X3=X1*COS(Q*PI/180)-Y1*SIN(Q*PI/180)
260 Y3=X1*SIN(Q*PI/180)+Y1*COS(Q*PI/180)
270 LINE (X1,T1)-(X3,T2),G
280 X1=X3:Y1=Y3
290 C=C-12
300 NEXT J
310 Y2=70
320 Y3=Y2*TAN(PI/3)-120
330 LINE (X1,Y1)-(X2,Y2),G
340 GLOCATE (X1+1,Y1+1),G+1,G
350 G=G+1
360 NEXT K

```





# START

## 色設定

D=12, Q=60  
L=30, N6, G=9  
S1=55, T1= S1\*TAN(25)

FORK=0 TO 14

もしも K<3なら N=6

N=N-1

もしも K>12なら N=5

R1=√L²-(L\*N/5)²

C=135

半径R1の円の135°の値X,Y

(X,Y) → (X1,Y1)

Q回転

(X1,Y1)と(S1,T1)を結ぶ

FOR J=1 TO 16

R1=0?

X2=R1\*cos(C)

Y2=R2\*sin(C)

(X2,Y2) → (X3,Y3)

Q回転

(X3,Y3)と(X1,Y1)を結ぶ

X1=X3, Y1=Y3

C=C-D

1

黄色系統の15階調

Dは角度の増分

Qは回転角

Lは山の大きさ

NはLを決めるパラメータ

R1は凸のパラメータ

値が大きければ大きいほど

盛りあがる。

初期設定時 C=135

R2=R1\*.6

補正值.6

突き出た部分を

描く

(135°-45°の角度の範囲)

## リスト3のプロチャート

## NEXT J

X2=55

X2=X2\*TAN(25)

(X1,Y1)と(X2,Y2)を

結ぶ

(160-K\*8,120)を

原点にとる

## 色の調整

NEKT K

END

非対称性の図形ができます。

135度から180度ひいた値と、135-(12\*15)=-45の関係から、おそらく賢い読者にはもうお分かりでしょう。

次に三角関数を利用して作った、「四プログラム」を2点紹介したいと思います。リスト4とリスト5を見てください。

自分でいうのも何ですが、まああの自信作です。

リスト4の行番号80の式は、高校時代に数学の先生から教わったものですが、ために軌跡を求めてみてください。

本プログラムは、この関数の特徴を生かして作ったものです。何とか、間が認識できるでしょう?

さて、次にリスト5です。これはリスト4のオマケみたいなものですが、とにかく紹介しておきます。リスト4と5の違いは、行番号70(リスト4)と行番号80(リスト5)、行番号80(リスト4)と行番号100(リスト5)です。できあがったものを見てみると、若干違ってきますね。

行番号80では、平方根の重みづけのため1.8倍し、行番号100では、20\*sin(H)-sin(3H)を用いています。

行番号100の関数の特徴は、Hが0~180のときZは正(90で最大)、180~360のときZは負(270で最小値をとる関数である点)を利用しています。

## リスト3

```
10 WIPE
20 GMODE 1
30 FOR I=1 TO 15
40 DEF PALETTE I,15,I,1
50 NEXT I
60 D=12:Q=60:L=30:N=6:G=9
70 GLOCATE (160,120),1,-1
80 FOR I=0 TO 14
90 IF I<3 THEN N=6
100 N=N-1
110 IF I>12 THEN N=5
120 R1=SQRT(L*L-(L*N/5)^2)
130 R2=R1*.6
140 S1=-55
150 T1=S1*TAN(25*PI/180)
160 C=135
170 X=R1*cos(C*PI/180)
180 Y=R2*sin(C*PI/180)
190 X1=X*cos(Q*PI/180)-Y*sin(Q*PI/180)
200 Y1=Y*cos(Q*PI/180)+X*sin(Q*PI/180)
210 C2=X1:T2=Y1
220 LINE (S1,T1)-(C2,T2),G
230 FOR J=1 TO 16
240 IF R1=0 THEN GOTO 330
250 X2=R1*cos(C*PI/180)
260 Y2=R2*sin(C*PI/180)
270 X3=X2*cos(Q*PI/180)-Y2*sin(Q*PI/180)
280 Y3=X2*sin(Q*PI/180)+Y2*cos(Q*PI/180)
290 LINE (X1,Y1)-(X3,Y3),G
300 X1=X3:Y1=Y3
310 C=C-D
320 NEXT J
330 X2=55
340 Y2=X2*TAN(PI*25/180)
350 LINE (X1,Y1)-(X2,Y2),G
360 GLOCATE (160+I*8,120)
370 IF G=7 OR G=14 THEN G=14:FLAG=1
380 IF FLAG=1 THEN G=G-1 ELSE G=C*1
390 NEXT I
400 END
```

## リスト4

```
20 WIPE:GLOCATE (160,120),1,-1
30 B=0:C=0:D=0:E=0
40 F=0
50 FOR A=80 TO 360 STEP 4
60 Y=(96+A*B)*.9
70 F=1.2*Y2
80 H=SQRT(A^2+Y^2)
90 Z=12*(COS(H*.2)-COS(H*.1))
100 IF A=360 THEN GOTO 330
110 LINE (X1,Y1)-(X2,Y2),G
120 X1=X2:Y1=Y2
130 NEXT A
140 B=B+3:C=C+5:E=E+10
150 IF E=160 THEN GOTO 200
160 GOTO 330
200 END
```



それでは  $\sin(H)$  と同じじゃないか、と思う読者がいるかもしれません。

実は  $\sin(H)$  でも同じ感じですが。しかし、いかにも、3Dをやっていると感じを出したかったのです。  $\sin(H)$  や  $\sin(3H)$  が奇関数(ある関数  $f(x)$  において、 $f(-x) = -f(x)$  が成り立つような関数のことをいう)であることを利用しました。

さて最後にリスト6です。

また、凹から凸への変化で、びっくりしないですね。

ここでは、行番号10の第2パラメータで、上方向を正にとっている点に注意してください。

行番号100~130で、ちょうど角度を0度から180度まで移動させています。

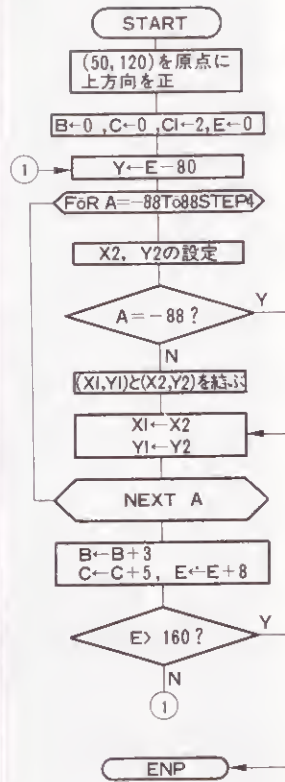
行番号200のEが山の高さのパラメータとなっており、行番号190と280のかね合いで、山が最高に達したら、今度は山の高さを小さくしていきます。

このプログラムは朝方のコケッコーを耳にしながら作りあげたもので、読者のなかには、おかしいと思う人がいるかもしれません。各自、どんどん直しちゃってください。一応、図1~図6に説明図、およびリスト1~6のフローチャートをのせておきます。参考にしてください。

それでは、また何かの記事で会えることを楽しみに。では、さようなら!! さようなら!!



図5 リスト4のフローチャート



:原点の設定

:初期設定  
色を緑に

$T = -2 * X2$

$H = \sqrt{A^2 + Y^2}$

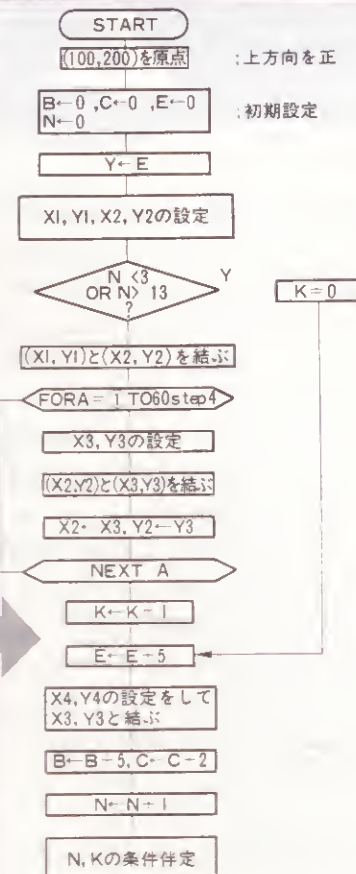
$Z = 12 * (\cos(H * 2) - \cos(H * 4))$

```

10 GMODE 1
20 WIFE:GLOCATE (50,140),1,-1
30 B=0:C=0:C1=5:E=0
40 Y=E-100
50 FOR A=-88 TO 88 STEP 4
60 X2=(96+A*B)*.9
70 T=-.2*X2
80 H=SQRT(A^2+Y^2)*1.8
90 H=H*PI/180
100 Z=20*SIN(H)-SIN(3*H)
110 Y2=T+Z+C
120 IF A=-88 THEN 140
130 LINE (X1,Y1)-(X2,Y2),C1
140 X1=X2:Y1=Y2
150 NEXT A
160 B=B+3:C=C+4:E=E+8
170 IF E>160 THEN 190
180 GOTO 40
190 END
  
```

(注) リスト5は上図の  $H = \sqrt{A^2 + Y^2} * 1.8$   
 $Z = 20 * \sin(H) - \sin(3H)$   
 $C = C + 4$  と変えるだけ

図6 リスト6のフローチャート



```

10 WIFE:GLOCATE (100,200),1,-1
20 B=0:C=0:E=0:N=0
30 Y=E
40 X1=85-B:X3=X1
50 Y1=X1+.2+C:Y3=Y1
60 X2=120-B
70 Y2=.2*X2+C
80 IF N<3 OR N>13 THEN E=0:GOTO 200
90 LINE (X1,Y1)-(X2,Y2),3
100 FOR A=0 TO 60 STEP 4
110 X3=120+A-E
120 T=-.2*X3
130 Z=Y*SIN(3*A*PI/180)
140 Y3=T+Z+C
150 IF A=0 THEN 170
160 LINE (X2,Y2)-(X3,Y3),3
170 X2=X3:Y2=Y3
180 NEXT A
190 K=K+1
200 E=E+5
210 X4=215-B
220 Y4=X4+.2+C
230 LINE (X3,Y3)-(X4,Y4),3
240 B=B+5:C=C+2
250 N=N+1
260 IF N=17 THEN END
270 IF N>13 THEN 40
280 IF K>5 THEN 300
290 GOTO 30
300 Y=Y-10
310 GOTO 40
  
```



# SMC ショートプログラム

ビンボの恋人  
ビンボの恋人

1

落合順子

こんにちは～落合順子です!!

アンケート葉書に、「彼になってあげますよ」とか「文通してください」と書いてくれた方々、ありがとう。こういう人たちはきっと、私と同じように SMC 大好き人間で、だからきつときつと、ゲームだけに熱中してるなんてことはないと思ってしまうのです。

それで今回は、いろーんな目的に SMC を使ってる人々のために、「ちょっと便利なツール集」です。私からのプレゼント♥

## ビンボから駒

私はビンボなので、プリンタを持っていません。そのくせ、いろんなプログラムを作ったり、グラフィックス・エディタで絵を描いたりするので、ファイルはごちゃっとあります。でもやはりビンボなので、無駄なディスクを買いたくない!! ということで、ファイルの整理を迫られてしまったのです。その上、自分でプログラムを開発している過程で、同じファイル名があっちこっちにあるのはバグのモト。ファイル名は同じなのに中身のバージョンが違う、とか、その逆とかね。

ファイルの整理をする場合、プリンタを持ってる人なら、ディレクトリをすべてプリンタに打ち出して、重複しているものなどをチェックするのもひとつの手。でも、同じファイル名とか探すの、けっこう大変。ましてや、プリンタがなかったらね。つまり、ファイル名の並べかえとか、検索とかができれば、うれしい。

でも…そんなソフト作るの、ちょっと大変だな…。だから既成のソフトをちょっとと利用。もちろんビンボだから買ったりしない。タダで付いてるソフト、MEMO を利用しましょ。


## ディスク・メモの利用法

まず、リスト1のBASICプログラムと、リスト2のマシン語パッケージを打ち込むこと。作ったパッケージファイルをリンクしてから、BASICのプログラムをRUNさせ

ます。

Insert disk please.と表示されたらディスクをセットしてリターンを押します。DISK NOと表示されたら、自分で決めた、そのディスクのナンバーを入力。次にディスクの名称(たとえば,"Basic work file")を入力。



すると、ファイル名とレコード数が表示されます。で、また次のディスクをセットして同じことの繰り返し。

ために2～3枚やってから、MEMO.EXEの入っている、空きのあるディスクをセットして、を押してください。そしてMEMOの見出しとなるインデックス名を入力。すると、MEMO??MEM(?)は数字)というファイルを作って終了します。

それでは Sony FILERに戻って、MEMOを起動しましょう。はーら、さっき

入力したインデックス名が表示されているでしょう。あとはsearchモードでソートして、重複などを見つけましょう。

## ☆賢い使い方☆


ファイル名でソートしたものを、いったんセーブします。そうしたら今度はEditモードで、いらないファイルや、他にコピーしたいファイルの色を変えます。かを押すと字の色が変わります。もう一度それをセーブして、またディスクナンバーでソートし直します。するとディスクごとになにを消したらいいか色で分かりますから便利です。

もちろんプリンタを持っている人は、ファイル名でソートした段階で、MEMOのプリント機能を使えばいいのです。

ちなみに、.MEMファイルを作った時に、他のMEMOファイルを消してしまったりすることはないので、安心してください。ただし、ディスクの枚数が多くなると、かなり大きなファイルになります。また、一度に扱えるディスクは30枚までで、1枚のディスクに50個以上のファイルがあると正しく動作しません。

## ビンボも楽しく

私はビンボなので、増設ドライブも持っていません。だからAREAコマンドってとっても不便なんです。ディスクの残り容量が知りたいときには、そのディスクに必ずAREAが入ってなきゃならないでしょ? たったの2Kだけでもつたないし、不便よね。それで、AREA.EXEを少し変更して、AREA2.EXEを作りました(リスト3)。

使い方や表示方法はAREAと同じです。ただ、AREAの場合、残り容量を表示した後、何かキーを押すとシステムに戻ってしまいますが、AREA2の場合、ここで他のディスクをセットして何かキーを押すと、そのディスクの残り容量を表示してくれます。つまり、AREA2がある1つのディスクに入っていれば、それを使って10枚でも20枚でも残り容量を調べることができます。システムに戻りたいときはキーを押してください。

## SMC SHORT PROGRAM

また、AREA.EXEは短くて分かりやすいプログラムなので、DEBUG.EXEで逆アセンブルして解析してみるとアセンブラの勉強になると思います。

\*\*\*変更部分\*\*\*

AREA.EXEの、02AB~02E4番地と01BF番地、0111番地を変更し、0400~0417番地をつけ加えます。ダンプリストでは0480~04FF番地もありますが、実際には必要ありません。

### ビンボでチェック

私はビンボなので……というのはどうでもいいんですが、やっぱりソフトって高いですね。その点、このOh! HitBitには、¥480の安さで良質なソフトがいっぱい!!これで打ち込む手間さえなければ……と思っている人も多いと思います。その上、マシン語にはチェックサムという便利なものがありますが、BASICにはなかったので、エラーにならない隠れたバグなど発見できず、大変不便だったと思います。

ところでMSXのBASICチェックサムはvol.4に掲載されましたが、とうとうSMC版もできました。しかもリンクパッケージなので、わずらわしいロード・セーブの繰り返しが必要ありません。つまり、BASICでBASICチェックサムプログラムを作ると、打ち込み中のプログラムをセーブする→チェックサムプログラムをロードする→誤りを発見する→打ち込み中のプログラムをロードする→誤りを直す……というようになるわけです。

ところがリンクパッケージにしておけば、ロードはまったくせずにすむわけです。さて、BASICチェックサムの仕様は、MSX版とまったく同じです。1行のうち、行番号を抜かした文字のASCIIコードをすべて加え、1つのサムとします。8行で横1行のサムとなります。

ただし、

- コメント文('の次に続く文字)はサムに加えない。この場合'もサムに加えません。('の代わりにREMを使ったものも同じ)。
- 空白(ASCIIコード20H)はサムに加えない。

### リスト1

```

1  ~~~~~
2  dump program      for SMC
3
4  please lri "sum.pac"
5
6  by JUN part II  .85. 2-7
7
8  ~~~~~
9  DIM XSUM(16) :CONSOLE SC :CCELEAR
10 INPUT "File name ":NAME$
11 FOPEN(NAME$)
12 ADR=H100
13 CCELEAR : LOCATE (0,2)
14 PRINT "Adr$; +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum"
15 PRINT STRING$(59,"-")
16 ALLSUM=0
17 FOR Y=1 TO 16
18   CCELEAR (Y MOD 2)*4+3
19   PRINT HEX$(ADR,4);": ";
20   ADR=ADR+H10 : YSUM=0
21   FOR X=1 TO 16
22     FINPUT A$
23     DUMP=ASC(A$)
24     PRINT HEX$(DUMP,2);": ";
25     YSUM=YSUM+DUMP
26     XSUM(X)=XSUM(X)+DUMP
27   NEXT
28   PRINT ":HEX$(YSUM,2)";
29 NEXT
30 CCELEAR 7
31 PRINT STRING$(59,"-")
32 PRINT "Sum : ";
33 FOR X=1 TO 16
34   PRINT HEX$(XSUM(X),2);": ";
35   ALLSUM=ALLSUM+XSUM(X)
36   XSUM(X)=0
37 NEXT
38 PRINT ":HEX$(ALLSUM,2)";
39 PRINT "Hit any key":B$=INKEY$(1)
40 IF ASC(MID$(A$,2,1))=0 THEN 50

```

●ただし、"に囲まれた文字列は空白もすべて加えます。この場合"自体は加えません。以上のことから、

1000 ' This is a program

1010 '

上の2行のサムは両方とも00です。

2000 PRINT A

2010 PRINT "A"

2020 PRINT "A

2030 PRINT A

上の4行のサムは同じですが、

2040 PRINT " A"

は異なります。

さて、次はどうでしょう。

3000 ' " "

3010 ' " "

3000行の方は00ですが、3010行の方は27H('のASCIIコード)となります。

\*\*\*リンクパッケージの使い方\*\*\*

CALL CHECK ("ファイル名.BAS")

でOKです。プログラムが長いときにはHit Any Keyで次のサムを表示します。すべて表示が終わると自動的にBASICのコマンド待ちに戻りますが、途中でもESCでブレイクできます。

\*\*\*おつきなおまけ\*\*\*

ところで、ひょっとして、まだマシン語のダンプ&チェックサムプログラムを持っていない人もいるのでは?

そこで、BASICチェックサムパッケージを利用して、マシン語のチェックサムプログラムを作りました。

リストを見れば分かりますが、BASICチェックサムパッケージに含まれている、FOPENとFINPUTというサブルーチンを呼んでいます。

FOPEN(文字列)はファイルをオープンします。そして、FINPUT(文字変数)で、ファイルからの1文字入力を行います。

たとえば、

CALL FINPUT (A\$)

とすると、もともとA\$に入っていた内容は無視して、2バイトの文字列が戻ってきます。1バイト目にファイルから入力された文字が、2バイト目にはヌル(ASCIIコード0の文字)が入っています。ファイルをすべて読み終わると1バイト目にヌル、2バイト目にはASCIIコード1AHの文字が入ります。

777-BASICには1文字入力がないので、何かと便利なオリエント……ではないでしょうか。



Addr:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	*	Ascii	Dump
0500:	44	20	81	01	01	10	04	20	40	88	88	88	80	80	00	00	:F3	D	.....	@1111
0510:	00	00	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	:6C		.....	
0520:	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	:A0		.....	
0530:	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	:A0		.....	
0540:	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	:A0		.....	
0550:	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	:A0		.....	
0560:	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	:A0		.....	
0570:	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A	:A0		.....	
0580:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	
0590:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	
05A0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	
05B0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	
05C0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	
05D0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	
05E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	
05F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00		.....	

# リスト 5

000	ORG 100H	0050	I	2090	DE=MSG1	7170	LINEEND
010	ENTRY FOPEN(STR)	0055	FLAG: DS 1	2100	JFRT	7180	A=(HL)
020	ENTRY FINPUT(STR)	0060		2110	C=16H	7190	#=AF
030	ENTRY CHECK(STR)	0065		2120	J0000H	71A0	#=HL
040	START	0070	INSUB:	2130	A=1BH	71B0	HL=YOI OSUM
050	HAZU: DS 1	0075	***	2140	MAIN2,00	71C0	A=A+(HL)
060		0080	A=(FLAG)	2150	JTRAF0	71D0	(HL)=A
070	FOPEN:	0085	A=12H	2160	OWARI:	71E0	HL=#
080	A=12H	0090	= NEXT,0	2170	DE=MSG2	71F0	AF=#
090	(FLAG)=A	0095	= EOF,0	2180	JFRT	7200	JASC
100	#=HL	0100	JFREAD	2190		7210	DE:HL
110	HL=005DH	0105	A=OFFH	2200		7220	A=A+(HL)
120	DE=005EH	0110	= EOF,0	2210	PRT:	7230	(HL)=A
130	BC=10	0115	A=0	2220	#=BC	7240	HL+
140	(HL)=20H	0120	NEXT:	2230	C=09H	7250	HL+
150	((DE+)=(HL+))	0125	HL=0080H	2240	J0005H	7260	JIN
160	HL=#	0130	B=00	2250	BC=#	7270	JUMF
170	A=(HL)	0135	C=A	2260		7280	HL+
180	A=C	0140	A+	2270		7290	HL+
190	= ER1,0	0145	(FLAG)=A	2280	SFLAG DS 1	7300	HL+
200	HL+	0150	HL=HL+BC	2290	WORD: DS 1	7310	JASC
210	E=(HL)	0155	A=(HL)	2300		7320	
220	HL+	0160	***	2310	GAL:	7330	JUMP: =JUN
230	D=(HL)	0165	EOF:	2320	HL=YOI OSUM	7340	
240	DE:HL	0170	A=12H	2330	(HL)=00H	7350	ASC:
250	DE=005DH	0175	(FLAG)=A	2340	HL=MOJI	7360	#=AF
260	B=A	0180	***	2350	DE=SUM	7370	A=2CH
270	A=1	0185	A=1AH	2360	B=6	7380	A=(HL)
280	SET:	0190		2370	L1:	7390	#=HL
290	#=AF	0195		2380	JIN	7400	JASC SUB
300	A=(HL)	0200	JINSUB	2390	A=15	7410	AF=#
310	A=OFFH	0205	A=1AH	2400	= NOTEOF,0	7420	HL+
320	= EX5,0	0210	I,0	2410	A=20H	7430	JASC SUB
330	A=C	0215	A=15	2420	NOTEOF:	7440	AF=#
340	A=1	0220		2430	(HL)=A	7450	HL+
350	= ER1,0	0225		2440	HL+	7460	ASC SUB: +7
360	C=1	0230		2450	= L1	7470	A=3AH
370	AI=#	0235	FWORI: DB 2	2460	B=B	7480	=0I,0
380	A=5	0240	FWORI: DW FSTR1	2470	JUN:	7490	A=A+07H
390	DE=0065H	0245	FSTR1: DS 1	2480	C=0	7500	OK:
400	= SETEND	0250	FSTR2: DS 1	2490	A=B	7510	(HL)=A
410	EX5:	0255		2500	A=B	7520	
420	A=7BH	0260	FINPUT:	2510	= EX,0	7530	
430	= ER1,0	0265	JINSUB	2520	JIN	7540	
440	A=61H	0270	DE=FWORI	2530	JIN	7550	wstat equ 0Ch
450	= EX2,0	0275	(FSTR1)=A	2540	JIN	7560	lofun equ 0Ch
460	A=A-0FH	0280	A=(FLAG)	2550	JIN	7570	intfl equ 0Ch
470	EX6:	0285	A=12H	2560	JIN	7580	fltin equ 0Ch
480	(DE)=A	0290	A=OFFH	2570	JIN	7590	retst equ 12h
490	AF=#	0295	FEND:	2580	EX:	7600	error equ 15h
500	DE+	0300	FEND:	2590	HL+	7610	trap equ 18h
510	SETEND	0305	FSTR2: CH	2600	HL+	7620	
520	A=1	0310	JRETSTR	2610	HL+	7630	japrom equ 0611ch
530	= ER1,0	0315		2620	LC:	7640	japvec equ 079b2ch
540	HL+	0320	CH=0	2630	JIN	7650	usarom equ 0d457h
550	A=0H	0325	CH=0	2640	A=5CH	7660	usavec equ 08000h
560	A=1	0330	CH=0	2650	= EX,0	7670	vector equ 00100h
570	HL+	0335	CH=0	2660	A=A+(HL)	7680	
580	SET	0340	CH=0	2670	(HL)=A	7690	wstat0: db wstat
590	A=0H	0345	CH=0	2680	JIN	7700	lofun0: db lofun
600	000SCH=0	0350	CH=0	2690	A=45H	7710	intfl: db intfl
610	000ABH=0	0355	CH=0	2700	= EX7,0	7720	fltin0: db fltin
620		0360	CH=0	2710	A=A+(HL)	7730	retst: db retst
630		0365	CH=0	2720	(HL)=A	7740	error: db error
640	DE=005CH	0370	CH=0	2730	JIN	7750	trap: db trap
650	10005H	0375	CH=0	2740	A=4DH	7760	
660	A=OFFH	0380	CH=0	2750	= EX7,0	7770	
670	= ER,0	0385	CH=0	2760	A=(HL)	7780	
680		0390	CH=0	2770	A=A+097H	7790	
690	ER:	0395	CH=0	2780	(HL)=A	7800	
700	A=10	0400	CH=0	2790	A=0E4H	7810	
710	JERROR	0405	CH=0	2800	= COMMENT	7820	
720	EP1:	0410	CH=0	2810	EX7:	7830	
730	A=13	0415	CH=0	2820	A=15	7840	
740	JERROR0	0420	CH=0	2830	= LINEEND,0	7850	
750		0425	CH=0	2840	A=59	7860	
760		0430	CH=0	2850	= EX1,0	7870	
770	KOUMOFU: DB 13,10,13,10,13,10	0435	CH=0	2860	(INPUT):	7880	
780	DB "Line: +0 +1 +2 "	0440	CH=0	2870	(WORD)=A	7890	
790	DB "+3 +4 +5 +6 +7:sum "	0445	CH=0	2880	A=C	7900	
800	DB "17,10,"\$ "	0450	CH=0	2890	A=0	7910	
810	FEI:	0455	CH=0	2900	A=(WORD)	7920	
820	DB "17,10,"\$ "	0460	CH=0	2910	= EX1,0	7930	
830	MOJI:	0465	CH=0	2920	EX2:	7940	
840	DB "17,10,"\$ "	0470	CH=0	2930	JIN	7950	
850	YOI OSUM:	0475	CH=0	2940	A=15	7960	
860	DB "17,10,"\$ "	0480	CH=0	2950	= LINEEND,0	7970	
870	SUM0:	0485	CH=0	2960	EX2	7980	
880	TATE SUM:	0490	CH=0	2970	EX1:	7990	
890	DB "17,10,"\$ "	0495	CH=0	2980	A=74	8000	
900	SUM:	0500	CH=0	2990	= EX7,0	8010	
910	DB "17,10,"\$ "	0505	CH=0	3000	A=12H	8020	
920	MSG1:	0510	CH=0	3010	A=0	8030	
930	DB "Line [ESC] +1,1 "	0515	CH=0	3020	C=4	8040	
940	MSG2:	0520	CH=0	3030	= L1	8050	
950	DB "End of file "	0525	CH=0	3040	EX7:	8060	
960	DB "Rev. 1.1,1 "	0530	CH=0	3050	A=20	8070	
970		0535	CH=0	3060	= EX4,0	8080	
980	FHEND:	0540	CH=0	3070	A=0	8090	
990	C=1AH	0545	CH=0	3080	= L1,0	8100	
1000	DE=0000H	0550	CH=0	3090	EX4:	8110	
1010	J0005H	0555	CH=0	3100	A=A+(HL)	8120	
1020	C=1AH	0560	CH=0	3110	(HL)=A		
1030	DE=0005H	0565	CH=0	3120	= L1		
1040	J0005H	0570	CH=0				



# Dr. LOGO

の世界

## 俵落しによる「学習」の第一歩

友里 翔

今回は「俵落しゲーム」を例にして、  
Logoにおける「学習」とは何か、  
という問題について考えていきたいと思います。  
その前に、前回出した宿題の解説をしておきましょうネ。

### 「あみだくじ」の解法

vol. 5では「あみだくじ」を作成するとこ  
ろまでを紹介しました。そして、実際に「あ  
みだ」をたどるプログラムを宿題として、  
読者の皆様からの反響に期待したのです。  
ところが、何としたことか、私の原稿のメ  
切日直前になっても、何の反応もないでは  
ありませんか。「ワシヤー怒ッタゾ」と手近  
にあったマイクロフロッピをむさぼり食お  
うとしていた矢先、編集者から天のヒト声。  
「解答が届きましたぞ」

というワケで、何とかメ切に間に合っ  
てしたのは、北海道は函館市にお住いのペン  
ネーム「森の木影」さん(26)です。それ  
では、さっそく内容を拝見しましょう(リ  
ストA)。

メインルーチンが solveamida ですね。これ

### リスト1

```
to solveamida
make "resultlist []
solve 1
setcursor [0 10]
show :resultlist
end

to solve :path
if :path > :path# [stop]
scan word char 64 + :path "0
solve :path + 1
end

to scan :key
local "nkey
if memberp :key :linklist
[make "nkey getlinked where
make "nkey
word first :nkey (bf :nkey) + 1]
[make "nkey search :key
if empty? :nkey
[make "resultlist
se :resultlist first :key
stop]]
scan :nkey
end

to search :key
if bf :key = :node# [op "]
local "nkey
make "nkey
word first :key (bf :key) + 1
if memberp :nkey :linklist
[op :nkey] [op search :nkey]
end

to getlinked :place
if even? :place
[op item :place - 1 :linklist]
[op item :place + 1 :linklist]
end

to even? :n
if remainder :n 2 = 0
[op "TRUE] [op "FALSE]
end
```

図1

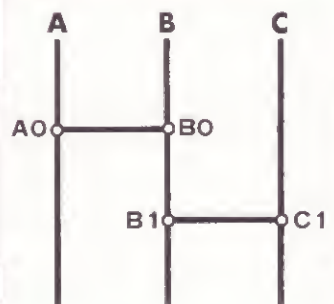


図2

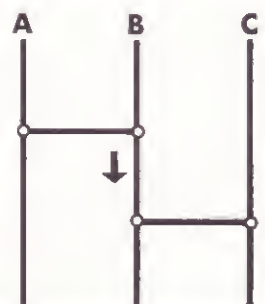


図3

A1, B1, A2, B2, A3の順で入れていって次にBが入れなくなったので、Bの負け。

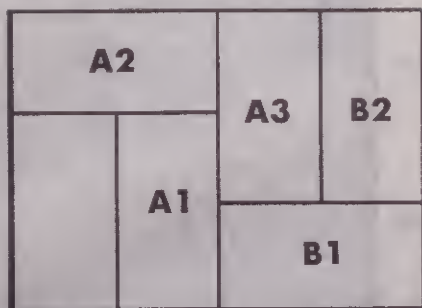
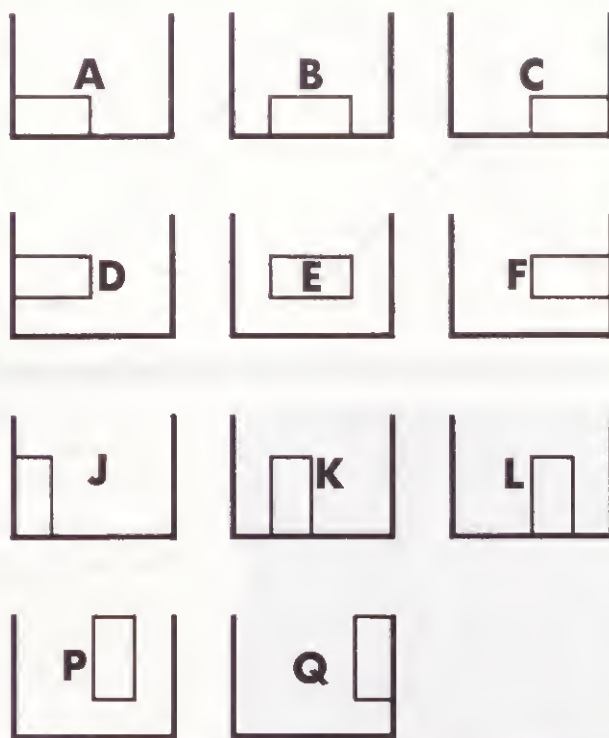


図4



はvol.5の「あみだくじ作成プログラム」に追加して使うと良いようです。resultlistという変数を初期化してsolveというプログラムを動かして、showしているところを見ると、結果はリストの型で表示されるようです。solveは各パスごとに道順をたどるプロシージャで、たとえばパスがAからDまであれば、A、B、C、Dの4つについてそれぞれの道順をたどります。solve 1の1はパスAを表します。

scanは、「あみだくじ」の横方向の接続を表すリストであるlinklistを実際にたどっていくプロシージャで、まず最初は引き数としてA0が与えられています。

どうやら、このscanが全体でもっとも重要な部分のようです。ここでは、getlinkedとsearchが使われています。getlinkedは引き数がリストの要素の場所を表しています。evenは引き数が偶数ならばTRUEを、奇数ならばFALSEを返す関数ですから、

getlinkedでは:placeが偶数ならばlinklist中の:place番目の要素のひとつ手前の要素を出力します。もちろん、奇数ならば:place番目の次の要素です。これは、どういうことでしょうか。

図1のあみだくじでは、linklistは[A0 B0 B1 C1]になります。ここでは、直接結ばれたノードは対のリストになっていますから、ノードA0に関するのはB0です。つまり、奇数番目の要素に関連する要素はその要素の次の要素、逆に偶数番目のC1に関連するノードはそのひとつ手前の要素B1ということになります。これでgetlinkedが関連のあるノードを出力するためのプロシージャだということが分かりました。

さて次はsearchです。どうやら、この引き数はノードの名前A0やB0などになっているようです。そうすると、なかの:nkeyは、そのノードのすぐ下のノードを指すことになります。もし:keyがB0なら、:

nkeyはB1になります。したがって、ここでは、ノードからリンクを伝って来た道順が次にパスを下って行くところをトレースしているのです(図2)。

これで、scanの内容は大体つかめられると思います。getlinkedで横方向へたどり、searchで縦方向へたどり、最後にたどり着いたパスの名前を:resultlistに追加していきます。この方式だと、図1の例では結果は[C A B]と出力されます。これは、AがCへ、BがAへ、CがBへとつながっていることを示しています。

「森の木影」さんのプログラムはインデントもきちんとしていてとても見やすいのですが、残念ながら結果がグラフィックで出力されていません。おいしいのですが、まあ良いでしょう。グラフィックスを導入するとすれば、scanあたりをもう工夫する必要があるそうですね。どうもありがとうございました。



図5

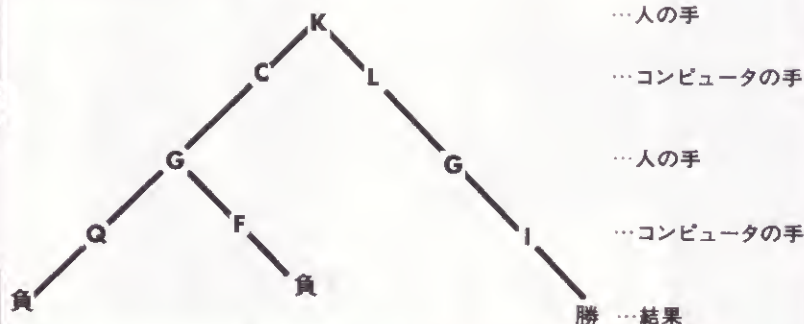
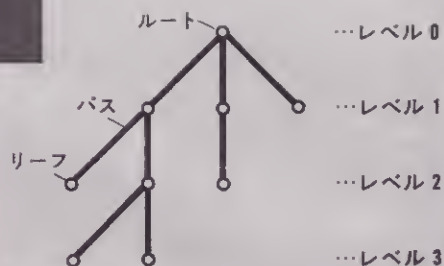


図6



## 俵落しゲーム

今回は、簡単なゲームを使ってコンピュータに学習させることを考えてみましょう。ただし、今回は学習部分については、メモリの制約上から実際上のプログラミングは行いませんでした。いつの日か全体をもう一工夫し、圧縮して何とかプログラムしてみたいところです。

### ■ルールの説明

「俵落し」のルールはきわめて簡単です。大きな箱に2人が順番につめ込んでいきます。最後に入れられなくなった方の負け。きちんと入れないと底の方にすき間ができしまい、本当は入るはずの俵も入らなくなってしまう。どうやって空間を作り先手後手を入れ替えるか、ここが大きなポイントになります(図3)。

ここでは、最終的にはコンピュータにやらせるという希望もあって、なるべく考えやすいように4×3の面を考えてみました。俵の大きさは2×1です。

### ■学習の第一歩

まず、人とコンピュータが戦うことにします。図3の勝負で、先手を人と考えますと、この例ではコンピュータが負けてしまいます。これを図4の俵の位置関係で示すと、K(人)→C(コ)→G(人)→Q(コ)→P(人)→負

の手順で負けたことが分かります。ここで着目すべき点は、Qを置いた時点で負けが決定的になったことです。ですから、ここではQが悪手なのです。

では、Qの代わりにFとしてみましょう。

K→C→G→Q→負

F→I→負

ところが、Fのあとに人にIを置かれて、やはり負けてしまいました。どうも、もっと前のCの手がいけないようです。これをLのように変えてみましょう。

K→C→G→Q→負

F→I→負

L→G→I→勝

この手順だと、人からGに置く場合、コンピュータはIに置けば勝てることになります。さて、先ほどから出ている手順図に注目してください。これはまるで、リストにしてください、といわんばかりのデータ構造をしているではありませんか。

人の第一手「K」に対して、コンピュータの手は何通りも考えられますが、それはデータ構造の空いている所を埋めていくことで求められます。どうやって埋めていくかについては、図4のAからQまでで箱に入

れることができるものを順に埋めていけば良さそうです。こうすると、このゲームでは人とコンピュータが対戦していて、始めはコンピュータが非常に弱く(始めはコンピュータ側に何の経験もないので、機械的に対応してくる)、何回かやっていくうちに少しずつ強くなる(先のデータ構造がどんどん完成に近づいていく…コンピュータは一度負けた手は2度と指さない)ということになります。

### ■木構造とリスト

さて、先に図5で示したようなデータ構造は、普通「木(tree)構造」と呼ばれています。ここでは、その「木」の各部の呼び方と、これをどのようにしてリストにするかをお話ししましょう。

「木」の一番上の部分(レベル0の部分)は、ルート(根)と呼ばれています。「枝葉」はここから下の方に向かって成長していきます。図6の白丸の部分はノードと呼ばれるものです。ですから、さっきのルートは正しくはルートのノードと呼ばれます。

「枝」はパス、一番下の方の、もうそれ以上パスの出ないノードはリーフ(葉)のノードと呼ばれています。

では、この木はいったいどうやってリストにしたら良いのでしょうか。図5の例を考えてみましょう。まずは左半分から。最初のゲームでは負けてしまいますが、K、C、G、Q、の順で手が進められました。これは素直にそのままリストにできそうであ

[K C G Q I]

最後のIは、lose(負)のことです。ここまでは簡単です。コンピュータはこの対戦で負けたことが分かっていますから、次からは2度とこの形からはQを指さないことになります。同じ途中経過を通った場合には、他の手を使います。ここではたまたまFを選びました。

[K C G F I]

これもまた、負けてしまいました。リストをこのようにいっぱい用意して記録をいっぱい増やすのもひとつの手ではありますが、前の2つのリストは内容が良く似ているので、これをひとつにしてみましょう。

[K C G [Q F] I]

これでも良いのですが、やはり図6でいうレベルに合わせて階層を持たせてやりましょう。

[K [C [G [Q [I] F [I]]]]]

この形だと、どんなに木が広がってもついていけそうです。図5の右半分を加えると、

[K [C [G [Q [I] F [I]]] L [G [I [W]]]]]

という形になります。

こうしてリストがどんどん大きくなるにつ

れ、コンピュータも経験をつんで少しずつ強くなっていきます。たとえば、コンピュータ内部でコンピュータ同士で対戦させておくと、コンピュータは勝手に勝負リストを作成していきますから、そのリストを持ったまま人間と対戦すると、いつの間にか強くなっている！ということになるのですね。いわゆる自己学習ですね。コンピュータは先のリストをたどりながら次の手を考えていますから、一度負けた手は絶対に打ってきません。まるで学習しているのと同じになっているのがお分かりいただけましたか。あとはメモリの大きさとの勝負ですが、左右対称な手はうまく一度で覚えるようにするなどの手法を使ったりして、データ圧縮をすることも可能です。

## ●実際のゲーム

理論的なことはだいたい理解していただけたと思うのですが、Dr. Logoではメモリがあまりにも少ないので、今回のものではまったく学習機能に手を出していません。人間同士で対戦するようになっています(黄組 vs 白組)。ただし、メインプロシージャである bagmain では黄・白共にまったく同じルーチンで組んでありますから、ゲーム

モリ的に何らかの余裕が生じた場合には、どちらかの getkey を computer とでも直して computer という名の学習プログラムに替えることもできます。

リストをすべて打ち込んだら bag②でゲーム開始です。ゲームは俵を落とすことによって進められますが、方法は2通りあり、「横にして落とす」と「縦にして落とす」のどちらかを選びます。ここでは選び方はカーソルキーを使っています。左キーが「横置き」で右キー「縦置き」。キーを押すたびに黒丸、白丸が画面上を動いて、どちらを選んでいるかが分かります。決めたらリターンキーを押してください(図7)。縦横を選ぶのは、プロシージャ selhv です。

リターンキーを押すと、今度は俵を置く位置を聞いてきます。横置きの場合、1から3までの数字が出るので、数字の位置に俵の左端があると思って選んでください。縦置きの場合には1から4まで、数字キーを押してください(図8)。これを実行しているのは、selnum です。

プログラム内部では、横置き1、2、3をそれぞれ h1、h2、h3 と、縦置き1、2、3、4を V1、V2、V3、V4 と呼んでいま

図7

1) 縦置きが選ばれている



2) 横置きが選ばれている



図8

Aは横置きで3を、Bは縦置きで4を選んだ例

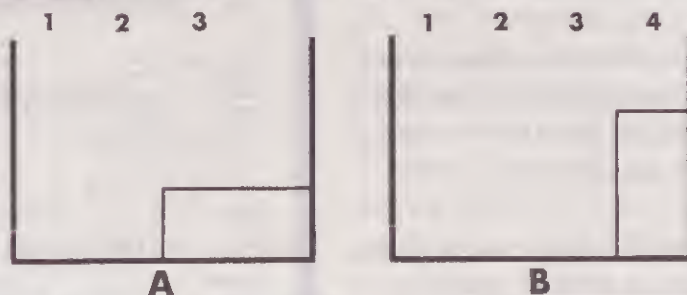
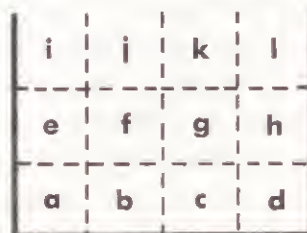


図9





す。また、これらは以前にどんなふうに変が入っていたかにもよりますが、図4の (A, D, P) (B, E, H) (C, F, I) (J, N) (K, O) (L, P) (M, Q) にそれぞれ対応しています。

ここでは箱の大きさを3×4に取っています。図9はこの箱の小部屋の呼び方を示しています。俵ひとつはこの小ブロック2個分の大きさなのです。

fillb では、実際にはそこに俵はないが上がふさがっていても新しく俵を入れることができないブロックを求めています。このなかのリスト [ieajf...] は3つずつで1組となっており、箱のブロックの縦割に対応しています。もしiに俵の一部があれば、e, a はもう使えませんが、fillb では[e a] が出力されます。

勝負がついたかどうかは winlose によってチェックされます。ここでは、h1, h2..... v3 v4 を順に入れてみて、エラーにならないかどうかをチェックしています。実際のチェックは chkbag で行われ、もしこれから入れようとする俵のための空間がない場合には FALSE を返してきます。実際に俵を入れるのは sinkbag によって行います。: bagmap という変数は、使われているブロックの登録用で、各プロシージャはこれを見ながら、箱の状態を知ることになります。

結果を画面に出力するのは hbag, vbag によります。fitpos は各俵の左上端の座標を出力します。fitpos には図4のA~Qを引き数として入力してください。

4×3の箱では人間がゲームするには少々もの足りないようですが、今後コンピュータといっしょにゲームをするという期待を残し、あえてコンパクトにしておきました。みなさんもこのプログラムを使ってみて、ここに示した方法以外の、新しいアルゴリズムによる「コンピュータ俵落とし」を考えてみてはいかがでしょうか。

## リスト2

```
to bag
  :- global variable -
  :: bagmap
  ;
  setsplit 25
  splitscreen
  ct
  ht
  makebox
  bagrule
  bagmapinit
  pu setpos [0 80] pd hbag
  pu setpos [70 80] pd vbag
  bagmain
end

to bagmain
  label "I"
  ;
  ; Yellow
  ;
  if winlose [Judgment "White stop"]
  setpc 6 pd
  putbag setkey
  setpc 7
  ;
  ; White
  ;
  if winlose [Judgment "Yellow stop"]
  setpc 7 pd
  putbag setkey
  go "I"
end

to makebox
  cs
  setpc 2
  pu setpos [-2 0] pd
  rt 180 fd 72 lt 90 fd 99 lt 90 fd 72
  rt 90 fd 10 rt 90 fd 82 rt 90 fd 119
  rt 90 fd 82 rt 90 fd 10
  setpc 7
end

to bagrule
  make "h1 [a b A e f D I J G]
  make "h2 [b c B f g E J k H]
  make "h3 [c d C g h F k I]
  make "v1 [e a J i e N]
  make "v2 [f b K j f O]
  make "v3 [g c L k g P]
  make "v4 [h d M i h Q]
end

to bagmapinit
  make "bagmap []
end

to drawbag :p
  fitpos :p
  if :p < "J [hbag] [vbag]
end

to fitpos :p
  (local "x "c)
  make "x ((ascii :p) - 64) * 2 - 1
  make "c piece :x :x + 1
  [0 2 1 2 2 2
   0 1 1 1 2 1
   0 0 1 0 2 0
   0 1 1 1 2 1 3 1
   0 0 1 0 2 0 3 0]
  make "x pen
  pu
  setpos se (first :c) * 24
             (first bf :c) * -24
  setpen :x
end

to hbag
  seth 0
  repeat 2 [rt 90 fd 46 rt 90 fd 22]
end

to vbag
  seth 0
  repeat 2 [rt 90 fd 22 rt 90 fd 46]
end

to getkey
  (local "sp "key)
  setcursor [20 6] pr [select bag]
  ;
  label "s"
  make "sp selhv "h
  ;
  label "I"
  make "key ascii rc
```

```
if not memberp :key [254 250 243]
[go "I]
[run item where [
[make "sp selhv "h go "I]
[make "sp selhv "v go "I]
[setcursor [27 6] pr [pos]
make "sp word :sp selnum :sp
if chkbag thing :sp [op :sp]
[setcursor [25 10]
pr "Error go "s]]]]
end

to selhv :sp
  local "n
  setcursor [20 10]
  repeat 8 [type "x "]
  if :sp = "h [make "n 0] [make "n 1]
  setcursor [22 3] pr char 17 + :n
  setcursor [32 3] pr char 18 - :n
  op :sp
end

to selnum :sp
  local "key
  make "key 0
  repeat 3
  [setcursor se 20 + :key * 4 10
   type char 49 + :key
   make "key :key + 1]
  if :sp = "v [setcursor [32 10] type "4]
  ;
  label "I"
  make "key (ascii rc) - 49
  if and :key < 3
  :key > -1 [go "m]
  if and :key = 3
  (:sp = "v) [go "m]
  go "I"
  ;
  label "m"
  setcursor se 20 + :key * 4 10
  pr char 17
  op :key + 1
end

to putbag :shp
  drawbag sinkbag thing :shp
end

to fillb :thishp
  eval memberp :thishp
  [i e a j f b k g c l h d]
  op piece
  where
  (((quotient (where - 1) 3) + 1) * 3)
  [i e a j f b k g c l h d]
end

to eval :n
end

to judgment :name
  setalpha 4
  setcursor [2 15]
  (pr :name [WIN !])
  setalpha 7
  setcursor [0 23]
end

to chkbag :thishp
  if emptyp :thishp [op "FALSE]
  if not or
  memberp first :thishp :bagmap
  memberp first bf :thishp :bagmap
  [op "TRUE]
  [op chkbag bf bf bf :thishp]
end

to chkbagloop :hv
  if emptyp :hv [op "FALSE]
  if chkbag thing first :hv
  [op "TRUE] [op chkbagloop bf :hv]
end

to sinkbag :thishp
  if not or
  memberp first :thishp :bagmap
  memberp first bf :thishp :bagmap
  [make "bagmap con :bagmap
   (se fillb first :thishp
    fillb first bf :thishp)
   op item 3 :thishp]
  [op sinkbag bf bf bf :thishp]
end

to winlose
  op not chkbagloop
  [h1 h2 h3 v1 v2 v3 v4]
end

to con :d :s
  if emptyp :s [op :d]
  if not memberp first :s :d
  [op con se :d first :s bf :s]
  [op con :d bf :s]
end
```

# SMC-C講座(4)

乗 松 保 智

黒 岩 方 人

北 堀 浩 紀

この講座も早いもので4回目となり、読者の皆様からの投稿も増えてきました。そこで、今回は講座を休講にして、投稿作品を御紹介いたしましょう。

## 黒岩さんのプログラムと質問

### その1.自作プログラムの投稿

#### 1) ファイルネーム

RECOVER.C (リスト1)

#### 2) プログラム作成のいきさつ

ある日、いつものようにフロッピー・マガジンのゲームなどを知人と楽しんだ後、知人を見送り、部屋に戻ると娘(小3)が「スピード・ゲームができないヨ」と言うのでした。「どれどれ」とどうせ使い方を間違えたのだらうと思いながら、ディスクをセットしメニュー(ゲームの)画面からスピード・ゲームを選択すると「Ready」を表示し、何も変化しないのです。「変だなー」とつぶやきながらSony FILERのシステムメニュー画面にしてみると、なんと確かあったはずの「SPEED.BAS」と「SPEED.PAC」が無いのです。

==現場検証の結果==

① command「ERASE」、filename「SPEED.XXX」を選ぶ。

②何やら「?(Y/N)」と聞いているので「Y」を押す(何かしたいときは必ずYとするから)。

③チットもゲームが始まらないので、もう一度①と②を行う。やっぱりゲームができない

という具合にスピード・ゲームは蒸発したのです。

ここで突然話題が変わるが、Sony FILERシステムメニューでの「ERASE」コマンドはキーボード入力にすべきだ!!そうしないと後世まで「ファイル蒸発の名器 Sony FILER」の汚名を語り継がれるゾ。話を戻して、サアーどうしよう、同一ソフトを再

度買うのもバカらしいので、ファイル復活のユーティリティが無いのかと捜すとCP/M上の物がありました。でもソワワもするのです。……とても買う気になれません……それじゃ自作でやってみるか。ということでプログラムの作成をしました。

## SMC-C

## STANDARD

## COURSE IV

### 3) プログラムの使い方

使用上難しいことは特に無し。ただし注意を要する。

A>RECOVER

drive no.1 (A) or 2 (B)? 1

track? 2

physical sector? 1

と入力するとSMC-777Cでは1番目のディレクトリのはいつているセクタ1の内容が表示される(図1参照)。

recover file exist? 1 (Yes) or 2 (No) 復活したいファイルが無ければ「2」を入力し、別のセクタを表示させる。セクタ番号は1→4→7→10→13→16→3→6→9→12→15→2→5→8→11→14の順

で行う。目的のファイルが見つかったら「1」を入力すると復活を行い、FCB先頭のデータはE5→00となり、セクタの内容が表示される。Ctrl-CでCP/Mに戻る。

### [使用上の注意]

「しまった、消した」と分かったら直ちにディスクまるごとコピーのバックアップディスクを作り、これがうまく復活できたら本番を行うこと。

### 4) プログラムの説明

bios(n, c)の関数を使いBIOSコールによりセクタダンプを行う。BIOSコールにより扱うセクタ論理セクタ(128バイト)であるので物理セクタ(256バイト)に変換する。ファイルの復活は1物理セクタ内にある8カ所すべてのFCBの先頭アドレスにデータ00を書き込むことにより行う。

### 5) プログラムの改善

1セクタの全ファイルに復活データを書き込む方法で投稿するのに気が引けましたが、下手なプログラムでちまちまと1データずつ変えるより、1セクタ「バッサリ」と変える方がスッキリしたやり方であると納得させて投稿することにしました。もし改善する場合は「FONT2.C」プログラムを参考に「スクリーン・エディタ方式」にしてみたいと思っています。

### 6) 参考文献

ナツメ社「パソコンユーザのためのCプログラミング技法」の「7.2-セクタダンププログラム」……この本のサンプルはすべてBDS-Cであるのでとても役に立つ。また編集も良い。

### その2.質問

1) 私はCのソース・プログラムの入力に「SMC EDIT」を使用していますが、アン



ダーライン (例, font 2.Cでの X\_\_OFS, Y\_\_OFS の"\_\_")が入力できません。"\_\_"を入力しようとすると"BS"と同じ動作をし1文字消えてしまいます。バグだと思うのですがいかがですか?

2) SMC-Cでシンボリック・デバッグを使おうと思ったのですが、ユーザーズ・マニュアルを読んでも「シンボリック・デバッグの作成について」という記事があり、必ずこの資料からお読みください」というので読んでも全く理解できない。おまけに「各マシンによって BDOS の位置が異なるため、一切お答えできませんので御了承ください。」と書いている。私のようにこれから「C言語」とは何だろうかと思って始める人達には何が何だか理解できないであろう。大体、SMC-70、SMC-77はRAM 64 Kが基本なのに60 Kの説明しても意味が無いのではないかと。結局何をどうすれば良いのか分からないので買ったときのまま使っているがこれが最適の状態であるのか? 以上の質問、よろしく御教示ください。

東京都 黒岩 方人 (40)

以上が黒岩さんのお便りですが、少々コメントしましょう。

#### その1. RECOVERについて

CP/MではFCBと呼ばれる36バイトのデータを用いてファイル管理を行っています。そのうち最後の4バイトを除いた32バイトがディレクトリ・エントリで、ディレクトリ・エリアはディレクトリ・エントリの集合であるわけです。ディレクトリ・エントリの最初の1バイトはそのファイルのユーザ・コード(0~15)を示しており、どのユーザにも所属していないディレクトリ(カラのディレクトリ)にはE5が書かれています。逆にいえば、ファイルを消すとはディレクトリ・エントリの第1バイトにE5を書き込むことを意味します。以上より、消してしまったファイルを復活するには該当ディレクトリ・エントリの第1バイトをE5→ユーザ・コード(通常は0)に書き換えれば良いわけです。

SMC-CP/Mの場合はブロック長は2 Kバイトですから、1つのディレクトリ・エントリで最大32 Kバイトのファイルを管理

できます。では32 Kバイトより大きいファイルはどうなるのかというと、複数のディレクトリ・エントリを1つのファイルが使用することになります。つまり、32 Kバイトを超えるファイルを復活するときには複数のディレクトリ・エントリに対してE5→0の書き換えを行わなければならないわけで、要注意です。CP/Mの内部構造に関して詳しくはアスキー出版局「応用CP/M」を参照してください。

さて、いよいよプログラムを見てみることにしましょう。まず目につくのがインデント(字下げ)のやり方です。もちろん、Cでは空白が書けるところにはいくつでもスペース、タブ、改行を書くことができるため、自由に書式を選ぶことができ、各人の好みの書式でプログラムを書くことができます。しかし、プログラムを人間が読むことを考

## SMC-C STANDARD COURSE IV

えると、読みやすい(理解やすい)書式というのはある程度限定されてきます。そのうちの1つがカーニハン&リッチーの「プログラミング言語C」の中でのサンプルプログラムの書式です。この本はCの文法書になっており、教科書ともいえるような本でありますから読者の皆様もこの書式に準じてプログラムを書くことをおすすめします。筆者もこの書式を採用しておりますので、Cプログラムを投稿して下さる方はできるだけこの書式でプログラムを書いてください。プログラムのブロック構造が直観的に見て取れるように書かれているものが美しいプログラムの1つの条件であると考えています。

リスト1でバグと思われる部分が1か所あります。main( )を見てください。whileループに初めて飛びこんだときにもし0以下

の数を入力したらどうなるでしょうか?そのときはwhileの条件を満足しないためループ本体は実行されずにsecread( )から実行されます。この時点でdrの値は適正か否かは分かりませんが代入されています。しかし、tr, secの値は代入されておらず、不定の値を持っています。この不定値を用いてデータの書き換えを行うことになってしまい、ぐあいが悪いのです。ではどうすれば良いかといいますと、その上のif文の中のbreakのところにて secread( )からの10行を移動させ、これらの文が実行されるときには、必ず変数の値が確定しているようにします。こうすれば、ドライブ番号の入力時に0以下の数を入力しても何もしないでプログラムが終了することになります。また、トラック、セクタ入力時に入力された値の妥当性のチェックがされていません。作成した人だけが使用するようなプログラムならともかく、投稿したりして数多くの人々に使われるようなプログラムではこのような(メンドクさいと思われるかもしれませんが)ことに注意するのも大切です。それから secread( )と secwrt( )ですが、ほとんど同じ処理で、リード ライトの方向がちがっているだけですから1つの処理にまとめてリスト2とし、呼び出し側も変更することによって、リード ライトを統一的にあつかうことができます。また、BIOS コールの機能番号ですが、リスト1では定数を直接使っていますが、リスト2では#defineで定義したシンボルを使っています。こうしておくほうがプログラムが読みやすくなりますね。BIOS、BDOS コールの機能コードを#defineでシンボル化したファイルを作り、BIOS、H、BDOS、Hと名付けておけば、これらのファンクション・コールを使用したければファイルの先頭で#includeすれば良く、後々便利だと思うのですがいかがでしょう。

#### その2. 質問に対する解答

1) 確かに SMC EDIT では通常はアンダーラインは入力できません。Cプログラムではよく使うキャラクタであるアンダーラインが入力できないなんて、SMC EDITにも困ったものですが、うまく入力する方法があります。アンダーラインをとりあえずマイナスなどに置き換えて入力しておき、

後で“X”コマンドでアンダーラインに変更すれば良いのです。

2) SMC-C のパッケージの中には CDB というシンボリック・デバッグが付属しています。CDB.COM と CDB 2.OVL がそれぞれです。CDB は、稼働する CP/M のサイズに依存しますので、最適なものを使用するためには再コンパイルが必要となります。供給されている実行可能な CDB パッケージは約 56 K 以上の CP/M で実行できますから 60 K である SMC-CP/M 上ではそのまま実行可能です。ただし最適の状態ではありません。CDB に関しては回を改めて説明したいと思います。

## 北堀さんの投稿

さて、続いては東京都の北堀さんの作品です。BASIC のグラフィック命令を C で実現したものです。リスト 3 が C で書いた部分、リスト 4 がアセンブリ言語で書いたプログラムを CRL ファイルに変換したダンプリストです。

以下に各関数の使用法を述べます。

```
circle (x, y, r, c, f)
int x, y, r, c, f;
中心 (x, y)、半径 r、色 c で円を描く。
f=0 が円、f=1 で塗りつぶす。

box (x, y, x1, y1, c, f)
int x, y, x1, y1, c, f;
(x, y) - (x1, y1) を対角線とする四角
を色 c で描く。f は circle と同様。

line (x, y, x1, y1, c, l)
int x, y, x1, y1, c, l;
点 (x, y) から (x1, y1) に直線を引く。
c は色、l は論理。
l=0: ノーマル
1: not
2: and
3: or
4: xor

int sin (a, x)      int cos (a, x)
int a, x;           int a, x;
a * sin (x) を返す。a * cos (x) を返す。
これらを使って三角関数を使った図形
を描くこともできる。

copy (n)
int n;
画面のハードコピー。プリンタに出力する
```

時に n によって縦倍率を指定することができる。

```
gclear (x)
char x;
グラフィック画面をデータ x でクリアする。

gmode (n)
int n;
グラフィックモードの設定。

gplot (x, y, c, l)
int x, y, c, l;
点 (x, y) に色 c でプロットする。l は論理
コード。

int gpoint (x, y)
int x, y;
点 (x, y) の色データを返す。

tmode (n)
int n;
```

# SMC-C STANDARD COURSE IV

表示画面の設定。

```
n=0: 40×25 ページ 0
1: 40×25 ページ 1
2: 80×25
```

```
cplot (x, y, c, n)
int x, y, n;
char c;
キャラクタ c を座標 (x, y) に出力する。
n は書き込む画面を示し、tmode ( ) の n
と同じ。

sound (l, h)
int l, h;
長さ l、高さ h の音を出す。

font (a, c)
int a;
char c [ ];
キャラクタ・フォントの定義。a がキャラク
タ・コード、C がそのデータの配列で、char
```

c[8]; と宣言しておく。a の b8 が 0 のと
き読み出し、1 のとき書き込みとなる。font
(0x41, c) で“A”のフォントが c に読み
出され、font (0x141, c) で C のデータが
41 にセットされる。

```
int cpoint (x, y, n)
int x, y, n;
座標 (x, y) の文字コードを下位 8 bit にア
トリビュートを上位 8 bit に返す。n はペー
ジ。

aclear (x, y, a, n)
座標 (x, y) のアトリビュートを a に設定
する。n はページ。

locate (x, y, n)
int x, y, n;
カーソルを座標 (x, y) に移動。

label (x, y, a, v, h, c, l)
int x, y, v, h, c, l;
char a;
グラフィック画面上の座標 (x, y) に縦
横 h 倍に色 c でキャラクタ a を表示する。
l は論理。

cclear (a, b, n)
int a, b, n;
ページ n をキャラクタ a、アトリビュート
b でクリア。

char inkey (n)
int n;
キー入力。getchar ( ) とはちがひ、n=1
で入力があるまで待つが、n=0 では待たな
い。アスキーコードを返す。

print (x, y, str, n)
int x, y, n;
char * str;
座標 (x, y) に str を出力。

wipe (a, n)
int a, n;
グラフィック画面の消去とテキスト画面を
a でクリアする。gclear ( ) と cclear ( )
をまとめたもの。
(以上ですが、紙面の関係上、アセンブリ
言語のソースを掲載できないことをおわび
します。)
```

## しぶとく続く C 講座

さて、今回は誌面の関係もありまして休講
とさせていただきます。次回からまた講
座を再開いたしますのでヨロシク。
御意見、御希望もヨロシク。



## リスト1

```

#define DMA 0x00B0
#define NULL 0

main()
{
    int dr, tr, sec, phy, find;
    find=0;
    while (1) {
        if (dr==0) {
            dr=0;
            tr=0;
            phy=0;
            sec=0;
            printf("track= %d sector= %d\n", tr, phy);
            printf("0 1 2 3 4 5 6 7 8 9 A B C D E F\n");
            secread(dr, tr, sec);
            dump(0);
            secwrite(dr, tr, sec+1);
            dump(8);
            printf("recover file exist 1(yes) or 2(no) ");
            if (find==1) break;
        }
        secread(dr, tr, sec);
        detachg();
        secwrite(dr, tr, sec);
        detachg();
        secwrite(dr, tr, sec+1);
        secread(dr, tr, sec);
        dump(0);
        secwrite(dr, tr, sec+1);
        dump(8);
    }

    input(name); /* INPUT statement of BASIC */
    char *name;

    {
        int p;
        printf(name);
        scanf("%d", &p);
        return(p);
    }

    secread(dr, tr, sec) /* 1 logical sector (128 byte) read */
    int dr, tr, sec;

    {
        bios(9, dr); /* set drive */
        bios(10, tr); /* set track */
        bios(11, sec); /* set logical sector */
        bios(12, DMA); /* set DMA */
        bios(13, NULL); /* 1 sector read to DMA */
    }

    secwrite(dr, tr, sec) /* 1 logical sector (128 byte) write */
    int dr, tr, sec;

    {
        bios(9, dr);
        bios(10, tr);
        bios(11, sec);
        bios(12, DMA);
        bios(14, NULL); /* 1 sector write from DMA */
    }

    dump(st) /* DMA memory dump */
    int st;

    {
        char *adr, d;
        int i, j;
        adr=DMA;
        for (j=0; j<16; j++) {
            printf("%02x ", (j+st)*16);
            for (i=0; i<16; i++) {
                printf("%02x ", adr[j*16+i]); /* hex dump */
                printf(" ");
                for (i=0; i<16; i++) {
                    d=adr[j*16+i];
                    if (d<0) printf(" ");
                    else printf("%c", d);
                }
                printf("\n");
            }
        }
    }

    detachg() /* ERA bit (E5) clear */

    {
        char *adr;
        for (adr=DMA; adr<DMA+0x80; adr=adr+0x20)
            *adr=0x00;
    }
}

```

### 図1セクタダンプ

```

track= 2 sector= 1
0 1 2 3 4 5 ..... F
                2 A .BIOS ..M...*
00 : 00 42 49 4F 53 20
10 :
    ERASEされると00→E5となる
    (注記)早い話が
F0 : これはセクタダンプ

```

## リスト3

```

int a, b, c, d, e, p, q;
char s[81];
b=0; c=0; d=0; e=0;
for (a=0; a<16; a++) {
    for (b=0; b<16; b++) {
        for (c=0; c<16; c++) {
            for (d=0; d<16; d++) {
                for (e=0; e<16; e++) {
                    gplot(p, y, d, f); --p;
                }
            }
        }
    }
}

circle(x, y, r, c, f)
int x, y, r, c, f;
{
    int a, b, l, ly, lx, ly1;
    bmode(0); b=0;
    if (b==0) { b=4; } else { b=2; }
    lx=sin(r, 0); ly=cos(r, 0)*b/5+y;
    for (a=0; a<360; a+=10) {
        lx1=sin(r, a); ly1=cos(r, a)*b/5+y;
        line(lx, ly, lx1, ly1, c, 0);
        lx=lx1; ly=ly1;
    }
    if (f!=0) {
        sss(x, y, -1, c); ++y;
        sss(x, y, 1, c);
    }
}

sss(x, y, z, c)
int x, y, z, c;
{
    int lx, ly, a;
    lx=1; ly=2;
    while (lx!=1) {
        a=-1; lx=shr(x, ly, c, a);
        a=1; lx=shr(x, ly, c, a);
        line(lx, ly, lx1, ly, c, 0); ly +=z;
    }
}

shr(lx, ly, c, a)
int lx, ly, c, a;
{
    int x;
    x=gpoint(lx, ly);
    if (x!=c) {
        while (x!=c) {
            lx += a;
            x=gpoint(lx, ly);
        }
    }
    else {
        return(1);
    }
}

copy(x)
int x;
{
    int a, b, c, d;
    prout(0x0d); prout(0x0a); prout(0x1b);
    prout(0x3e); prout(0x1b); prout(0x54);
    if (x%8) x=8;
    dmode(0); d=0; a=640; b=0x36; c=0x34;
    if (d==0) { a=320; b=0x33; c=0x32; }
    if (x/2) copy0(a, b, c);
    else { copy1(a, b, c); }
    prout(0x1b); prout(0x3e);
    prout(0x1b); prout(0x3e);
}

copy1(a, m1, m2, r)
int a, m1, m2;
{
    int b, c, d, e, p, y, z;
    b=0x30; c=x;
    if (x/4) { b=0x31; c=-5; } c=1; c!=0x30;
    prout(b); prout(c);
    for (d=0; d<200; d++) {
        prout(0x1b); prout(0x53); prout(0x30);
        prout(m1); prout(m2); prout(0x30);
        for (b=0; b<16; b++) {
            p=0; d=gpoint(b, c);
            if (d!=0) d=1;
            p += d;
            for (e=0; e<16; e++) {
                p = 1 + p + d;
                prout(p);
            }
            prout(0x0d); prout(0x3e);
        }
    }
}

copy0(m1, m2)
int m1, m2;
{
    int a, b, c, d, e;
    prout(0x31); prout(0x30);
    for (a=0; a<25; a++) {
        c=0x8;
        prout(0x1b); prout(0x53); prout(0x30);
    }
}

```

## リスト2

```

#define SELDRV 9
#define SECREAD 13
#define SECWRT 14

void secrw(rwmode)
int rwmode;
{
    bios(SELDREV, dr);
    bios(SECREAD, tr);
    bios(rwmode, NULL);
}

void main()
{
    secrw(SECREAD);
    secrw(SECWRT);
}

```

1.59



# Oh!HITBIT原稿募集

当編集部では、読者の皆さまからの自作プログラム、原稿を募集いたします。ジャンルは、次の3つです。

①自作プログラム

ゲーム、ビジネス、グラフィック、ユーティリティ、ツールなど、SMCシリーズあるいはHBシリーズで使えるプログラム、気軽に打ち込めるショート・プログラムを送ってください。CP/MなどのOS上で動くプログラムでもけっこうです。

②ハード製作

③Oh!HITBIT 読者にふさわしい読みものSF、ショート・ショート、論文、またはぐっとくだけで「わが家ではHITBITをこんなふうに使っている」というレポートも大歓迎。

応募要領

原稿には、住所、氏名（ペンネーム希望の方はペンネームも）、年齢、連絡先電話番号、職業、そして対象機種、必要な周辺機器を必ずご記入ください。

①自作プログラム

プログラムをセーブしたテープ、ディスクでご応募ください。原稿は原則としてお返しませんが、ディスクは返送希望と書いていただければ、返送いたします。

Oh!HITBITは、初心者も対象にしていますので、できるだけ分かりやすい内容の説明をお願いします。

フローチャート、変数表、メモリマップなども、できるだけ添えてください。

リストは、プリンタのない方はプログラム

メディアのみの応募でけっこうですが、同封を原則とします。マシン語の場合、ダンプリストのほかに、ソースリストも必ずつけてください。

また、参考文献、引用資料などは明記してください。

②ハード製作

製作したハードをお送りいただかなくても図面、原稿などを検討の上、当編集部より連絡いたします。

回路図は、大きく鮮明に描いてください。トレースの必要はありませんが、部品名、ピン番号などを見やすく、スペースをとって書くようお願いします。また、実体配線図、部品表もつけてください。

掲載方法

毎号、優秀作品をOh!HITBIT誌上に掲載します。原稿は、掲載にあたり修正させていただきます場合があります。あらかじめご了承ください。

掲載作品には、本誌規定の原稿料をさし上げます。

9月18日発売予定

次号予告

最近、読者の方からの投稿が増えてきて、編集者一同キャアキャアと喜んでおります。いやー、Oh!HITBITの読者はレベルが高いぜ!! SMCの原稿が圧倒的に多いようですが、MSXのユーザーもがんばってね。プログラムに限らず、「私の使い方」「ゲームの攻略法」「ワタシャさっぱりわからない」なんでもよござんす。くじけず手紙を書いてください。とゆーところで凸版印刷の菊やん登場。「もう時間がありません。失禁しそうですよ」。失禁されてはたまらないのでこの原稿を渡さなくては。次号は、ゲームについて考えてみたいですね。菊「ピース」

Oh!HITBIT Vol.6 480円

1985年7月1日発行

- 発行人 孫正義
- 編集人 田鎖洋治郎
- 発売元 株式会社 日本ソフトバンク  
出版部 〒102 東京都千代田区四番町2-1  
TEL03(261)4095(代)
- 本社 〒102 東京都千代田区九段南2-3-14  
TEL03(263)3690(代)
- 大阪支店 〒542 大阪市南区難波千日前5-19  
TEL06(644)0191(代)
- 印刷 凸版印刷株式会社
- 編集 西澤伸樹
- 技術 松田辰夫 徳永聡
- 海外協力 駒形成美
- 協力 榎本博行 西田明宏 乗松保智
- カメラ 浜崎昭 杉山和美
- ©1984SOFTBANK雑誌02281-7  
本誌からの無断転載を禁じます。
- COVER: Yuriko Arata
- DESIGN: Shoko Fujiyama

# SONY



## 【マイクロフロッピーをロードすれば、 一歩進んだ自分に出会える。】



### 「僕らMFD世代。」

自らの頭脳とセンスでキーボードをたたき、CGやプログラムをつくる。自分の世界をクリエイトするために、3.5インチをロードする。そんなMFD世代がいまいちばん生き生きしています。マイクロフロッピーディスクは、独自のハイプレジジョンVIVAX磁性体を採用。データエラーにつながるノイズを押さえ、高S/N比、高出力、高分解能を実現しています。パソコンがカラダの一部になってしまった僕らMFD世代のメディアです。

MFD-1 片面0.5Mバイト ¥1,350 MFD-2 両面1.0Mバイト ¥1,750

マイクロフロッピーディスク

## ソニーの3.5インチMFD

基本性能(MFD-1・2共通) ●トラック数：80トラック(サイド)、40トラック(サイド) ●トラック密度：135TPI(80トラック)、67.5TPI(40トラック) ●記録密度：8,187BPI(サイド"0"), 8,717BPI(サイド"1") ●変調方式：MFM ●記憶容量：0.5Mバイト(MFD-1)、1.0Mバイト(MFD-2) ●転送速度：500kビット/秒(600RPM)



# SONY

バンゲリングベイ帝国の全貌がつかめたら本物だ。



あのロードランナーをつくった、ブローダーバンド社の大作、バンゲリングベイ。MSXになって、ソニーから誕生。



**バンゲリングベイ ¥4,900**  
HBS-G036C (ROM版) (16Kバイト以上)  
©1984 Broderbund Software, Inc. by Will Wright  
7月21日発売予定。予約◎の大人気だ！

**MSX** 世界征服の野望を抱いてきたバンゲリング帝国は、ついに最終兵器を完成させようとしていた。君の使命は、特殊武装ジェットヘリを駆使して帝国の秘密兵器工場を破壊することにある。ところで、画面に現れるのは、全体の約100ここで君の記憶プラス構成力をもって帝国全体のようすをつかまなければ、迷い子になってしまう。このジェットヘリには、工場や戦艦を破壊する爆弾と、敵戦闘機やレーダーなどを攻撃するバルカン砲がある。戦略としては、まず敵の重要拠点の工場をたたくこと。高得点をあげると、名誉の勲章が画面に現れる。では、一流戦士としての君のテクニックを待っている。

**HIT BIT MEZZO**

RAM64Kバイト搭載。ソフトが気軽に幅広く楽しめる。新しいイメージ。●写真はHB-201本体 ¥55,800とトリニトロンカラーテレビ ¥145,800 ¥56,000との組み合わせです。



ひとびとのヒットビット。

**HIT BIT**

●メーカー名、型番、価格、電話番号、機種名を明記の上、〒105 東京都港区  
最上町一丁目一番一丁目までお申し込みください。☐はマイクロソフト社の商標です。  
●☐はMicrosoftの登録商標です。RAM容量以上のパソコンシステムでお使いください。